



UNIVERSIDAD DE GUANAJUATO

CAMPUS IRAPUATO - SALAMANCA

DIVISIÓN DE INGENIERÍAS

*“Modelo de detección de enfermedades pulmonares
en imágenes de rayos X por medio de inteligencia artificial”*

TESIS

PARA OBTENER EL GRADO DE:

MAESTRÍA EN INGENIERÍA ELÉCTRICA

PRESENTA:

Ing. Maria Camila Isaza Lancheros

DIRECTORES:

Dr. Juan Gabriel Aviña Cervantes

Dr. Cesar Hernando Valencia Niño

Salamanca, Guanajuato

Octubre 2024

Salamanca, Gto., a 08 de noviembre del 2024.

M. en I. HERIBERTO GUTIÉRREZ MARTIN
COORDINADOR DE ASUNTOS ESCOLARES
P R E S E N T E.-

Por medio de la presente, se otorga autorización para proceder a los trámites de impresión, empastado de tesis y titulación al alumno(a) **MARIA CAMILA ISAZA LANCHEROS** del **Programa de Maestría en INGENIERÍA ELÉCTRICA** y cuyo número de **NUA** es: **147870** del cual soy director. El título de la tesis es: **"Modelo de detección de enfermedades pulmonares en imágenes de rayos X por medio de inteligencia artificial"**

Hago constar que he revisado dicho trabajo y he tenido comunicación con los sinodales asignados para la revisión de la tesis, por lo que no hay impedimento alguno para fijar la fecha de examen de titulación.

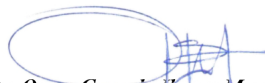
ATENTAMENTE



Dr. Juan Gabriel Aviña Cervantes
DIRECTOR DE TESIS
SECRETARIO



Dr. César Hernando Valencia Niño
DIRECTOR DE TESIS



Dr. Oscar Gerardo Ibarra Manzano
PRESIDENTE



Dr. Juan Pablo Ignacio Ramírez Paredes
VOCAL

Dedicatoria

A mis padres Sra. Bertha Lancheros y Sr. Francisco Isaza.

Ing. Maria Camila Isaza Lancheros

Agradecimientos

Al Dr. Gabriel Aviña por su apoyo, orientación, dirección y formación durante la maestría.

Al Dr. César Valencia por su confianza, apoyo, orientación y dirección, siendo pilar importante en mi desarrollo académico y profesional.

A la Ing. Leidy Retamozo Triana por todo su conocimiento y ayuda incondicional en la culminación de este proyecto.

A mi prima Alejandra Isaza, por hacer de mis días mas llevaderos y alegrarme la vida a lo largo de este proceso.

A todas aquellas personas que de una u otra manera hicieron posible la realización de este trabajo de Tesis.

Ing. Maria Camila Isaza Lancheros

Agradecimientos Institucionales

Expreso mi más sincera gratitud hacia la Universidad de Guanajuato, especialmente a la División de Ingenierías del Campus Irapuato-Salamanca (DICIS) por la formación académica y por el apoyo financiero que he recibido durante mis estudios en esta institución.



A mi Alma Mater, la Universidad Santo Tomas Seccional Bucaramanga, Colombia, base de mi sólida formación académica.



UNIVERSIDAD SANTO TOMAS
PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA
BUARAMANGA

Este trabajo de tesis fue realizado gracias al apoyo invaluable recibido a través del Consejo Nacional de Ciencia y Tecnología CONACYT de México, bajo el número de CVU 1103477.



Ing. Maria Camila Isaza Lancheros

Índice General

Dedicatoria	ii
Agradecimientos	iii
Agradecimientos Institucionales	iv
Índice de Figuras	ix
Índice de Tablas	x
Resumen	xvii
Abstract	xvii
1 Introducción	1
1.1 Introducción	1
1.2 Antecedentes	2
1.3 Justificación	3
1.4 Objetivos	4
1.4.1 Objetivo general	4

1.4.2	Objetivos específicos	4
1.5	Planteamiento general del problema	4
1.6	Organización de la tesis	5
2	Marco Teórico	6
2.1	Conceptos de Redes Neuronales	7
2.2	Funciones de Activación	7
2.2.1	Función Sigmoide	7
2.2.2	Función ReLu (Rectificador Linear Unitario)	8
2.2.3	Función Leaky ReLu	8
2.2.4	Función SoftMax	9
2.3	Aprendizaje	9
2.3.1	Procesamiento de las imágenes	9
2.3.2	Convolución	11
2.3.3	Reducción/Pooling	11
2.3.4	Normalización del Batch (BatchNormalization)	12
2.3.5	Dropout	13
2.4	Función de pérdidas o coste	13
2.4.1	Error medio cuadrático	14
2.4.2	Error medio absoluto	14
2.4.3	Entropía cruzada categórica (Categorical Cross-Entropy)	14
2.4.4	Entropía cruzada binaria (Binary Cross-Entropy)	14
2.5	Optimizadores de función de pérdidas o coste	15
2.5.1	Descenso estocástico del gradiente (SGD)	15
2.5.2	Método adaptativo (Adagrad)	16
2.5.3	RMSprop	16
2.5.4	Estimación Adaptativa de Momentos (Adam)	17
2.5.5	Estimación del Momento Adaptativo acelerada por Nesterov (Nadam)	17

2.6	Kernel de Inicialización	18
2.6.1	GLOROT	18
2.6.2	HE	18
2.7	Métricas de Validación	19
2.7.1	Matriz de Confusión	19
2.7.2	Precisión	19
2.7.3	Sensibilidad (Recall)	20
2.7.4	Especificidad	20
2.7.5	Exactitud (Accuracy)	20
2.7.6	F1 Score	20
3	Pruebas de Arquitecturas	21
3.1	Introducción	21
3.2	Arquitecturas Libres	22
3.2.1	Selección 70/30	22
3.2.2	Selección 80/20	28
3.2.3	Selección 95/5	34
3.3	Arquitecturas Conocidas	40
3.3.1	Selección 70/30	41
3.3.2	Selección 80/20	46
3.3.3	Selección 95/5	51
3.4	Filtro 1: Modificación de la función de Activación	56
3.5	Filtro 2: Modificación del optimizador	57
3.6	Filtro 3: Modificación Kernel de Inicialización	57
4	Conclusiones	58
	Referencias	60

Índice de Figuras

1.1	Organización de un perceptrón por Frank Rosenblatt.	2
1.2	La primera radiografía de la historia: la mano de Anna Bertha Ludwig.	3
1.3	Estructura de una red neuronal convolucional (CNN)	4
2.1	Estructura de una red neuronal.	8
2.2	Proceso de Binarización de una imagen de rayos X	10
2.3	Diferentes tipos de Segmentación	10
2.4	Ejemplo de normalización de una imagen.	11
2.5	Diagrama de extracción de características mediante una CNN	11
2.6	Ejemplo de Convolución con función de activación ReLu	12
2.7	Ejemplo de los métodos de Pooling	12
2.8	Red antes y después de aplicar Dropout	13
2.9	Ejemplo de SGD	15

Índice de Tablas

2.1	Matriz de Confusión	19
3.1	Estructura de la Red Neuronal Convolutiva Libre 1.	22
3.2	Hiperparámetros de la Red Neuronal Convolutiva Libre 1	22
3.3	Matriz de Confusión Arquitectura Propuesta 1	23
3.4	Porcentaje de error y exactitud Arquitectura Propuesta 1	23
3.5	Métricas Arquitectura Propuesta 1	23
3.6	Estructura de la Red Neuronal Convolutiva Propuesta 2	23
3.7	Hiperparámetros de la Red Neuronal Convolutiva Propuesta 2	24
3.8	Matriz de Confusión Arquitectura Propuesta 2	24
3.9	Porcentaje de Error y exactitud Arquitectura Propuesta 2	24
3.10	Métricas Arquitectura Propuesta 2	24
3.11	Estructura de la Red Neuronal Convolutiva Propuesta 3	25
3.12	Hiperparámetros de la Red Neuronal Convolutiva Propuesta 3	25
3.13	Matriz de Confusión Arquitectura Propuesta 3	25
3.14	Porcentaje de Error y Exactitud Arquitectura Propuesta 3	25
3.15	Métricas Arquitectura Propuesta 3	26

3.16 Estructura de la Red Neuronal Convolutacional Propuesta 4	26
3.17 Hiperparámetros de la Red Neuronal Convolutacional Propuesta 4	26
3.18 Matriz de Confusión Arquitectura Propuesta 4	26
3.19 Porcentaje de Error y exactitud Arquitectura Propuesta 4	27
3.20 Métricas Arquitectura Propuesta 4	27
3.21 Estructura de la Red Neuronal Convolutacional Propuesta 5	27
3.22 Hiperparámetros de la Red Neuronal Convolutacional Propuesta 5	27
3.23 Matriz de Confusión Arquitectura Propuesta 5	28
3.24 Porcentaje de Error y exactitud Arquitectura Propuesta 5	28
3.25 Métricas Arquitectura Propuesta 5	28
3.26 Estructura de la Red Neuronal Convolutacional Libre 1.	28
3.27 Hiperparámetros de la Red Neuronal Convolutacional Libre 1	29
3.28 Matriz de Confusión Arquitectura Propuesta 1	29
3.29 Porcentaje de error y exactitud Arquitectura Propuesta 1	29
3.30 Métricas Arquitectura Propuesta 1	29
3.31 Estructura de la Red Neuronal Convolutacional Propuesta 2	29
3.32 Hiperparámetros de la Red Neuronal Convolutacional Propuesta 2	30
3.33 Matriz de Confusión Arquitectura Propuesta 2	30
3.34 Porcentaje de Error y exactitud Arquitectura Propuesta 2	30
3.35 Métricas Arquitectura Propuesta 2	30
3.36 Estructura de la Red Neuronal Convolutacional Propuesta 3	31
3.37 Hiperparámetros de la Red Neuronal Convolutacional Propuesta 3	31
3.38 Matriz de Confusión Arquitectura Propuesta 3	31
3.39 Porcentaje de Error y Exactitud Arquitectura Propuesta 3	31
3.40 Métricas Arquitectura Propuesta 3	32
3.41 Estructura de la Red Neuronal Convolutacional Propuesta 4	32
3.42 Hiperparámetros de la Red Neuronal Convolutacional "Propuesta 4	32

3.43 Matriz de Confusión Arquitectura Propuesta 4	32
3.44 Porcentaje de Error y exactitud Arquitectura Propuesta 4	33
3.45 Métricas Arquitectura Propuesta 4	33
3.46 Estructura de la Red Neuronal Convolutacional Propuesta 5	33
3.47 Hiperparámetros de la Red Neuronal Convolutacional Propuesta 5	33
3.48 Matriz de Confusión Arquitectura Propuesta 5	34
3.49 Porcentaje de Error y exactitud Arquitectura Propuesta 5	34
3.50 Métricas Arquitectura Propuesta 5	34
3.51 Estructura de la Red Neuronal Convolutacional Propuesta 1.	34
3.52 Hiperparámetros de la Red Neuronal Convolutacional Propuesta 1	35
3.53 Matriz de Confusión Arquitectura Propuesta 1	35
3.54 Porcentaje de error y exactitud Arquitectura Propuesta 1	35
3.55 Métricas Arquitectura Propuesta 1	35
3.56 Estructura de la Red Neuronal Convolutacional Propuesta 2	35
3.57 Hiperparámetros de la Red Neuronal Convolutacional Propuesta 2	36
3.58 Matriz de Confusión Arquitectura Propuesta 2	36
3.59 Porcentaje de Error y exactitud Arquitectura Propuesta 2	36
3.60 Métricas Arquitectura Propuesta 2	36
3.61 Estructura de la Red Neuronal Convolutacional Propuesta 3	37
3.62 Hiperparámetros de la Red Neuronal Convolutacional Propuesta 3	37
3.63 Matriz de Confusión Arquitectura Propuesta 3	37
3.64 Porcentaje de Error y Exactitud Arquitectura Propuesta 3	37
3.65 Métricas Arquitectura Propuesta 3	38
3.66 Estructura de la Red Neuronal Convolutacional Propuesta 4	38
3.67 Hiperparámetros de la Red Neuronal Convolutacional Propuesta 4	38
3.68 Matriz de Confusión Arquitectura Propuesta 4	38
3.69 Porcentaje de Error y exactitud Arquitectura Propuesta 4	39

3.70 Métricas Arquitectura Propuesta 4	39
3.71 Estructura de la Red Neuronal Convolutacional Propuesta 5	39
3.72 Hiperparámetros de la Red Neuronal Convolutacional Propuesta 5	39
3.73 Matriz de Confusión Arquitectura Propuesta 5	40
3.74 Porcentaje de Error y exactitud Arquitectura Propuesta 5	40
3.75 Métricas Arquitectura Propuesta 5	40
3.76 Estructura de la Red Neuronal Convolutacional LeNet 1	41
3.77 Hiperparámetros de la Red Neuronal Convolutacional LeNet 1	41
3.78 Matriz de Confusión Arquitectura LeNet 1	41
3.79 Porcentaje de Error y exactitud Arquitectura LeNet 1	41
3.80 Métricas Arquitectura LeNet 1	42
3.81 Estructura de la Red Neuronal Convolutacional LeNet 1 con Variación	42
3.82 Hiperparámetros de la Red Neuronal Convolutacional LeNet 1 con Variación	42
3.83 Matriz de Confusión Arquitectura LeNet 1.2	42
3.84 Porcentaje de Error y exactitud Arquitectura LeNet 1 con Variación	43
3.85 Métricas Arquitectura LeNet 1 con Variación	43
3.86 Estructura de la Red Neuronal Convolutacional LeNet 5	43
3.87 Hiperparámetros de la Red Neuronal Convolutacional LeNet 1 con Variación	43
3.88 Matriz de Confusión Arquitectura LeNet 5	44
3.89 Porcentaje de Error y exactitud Arquitectura LeNet 5	44
3.90 Métricas Arquitectura LeNet 5	44
3.91 Estructura de la Red Neuronal Convolutacional LeNet 5 con Variación	44
3.92 Hiperparámetros de la Red Neuronal Convolutacional LeNet 1 con Variación	45
3.93 Matriz de Confusión Arquitectura LeNet 5.2	45
3.94 Porcentaje de Error y exactitud Arquitectura LeNet 5.2	45
3.95 Métricas Arquitectura LeNet 5 con Variación	45
3.96 Estructura de la Red Neuronal Convolutacional LeNet 1	46

3.97 Hiperparámetros de la Red Neuronal Convolutacional LeNet 1	46
3.98 Matriz de Confusión Arquitectura LeNet 1	46
3.99 Porcentaje de Error y exactitud Arquitectura LeNet 1	46
3.100 Métricas Arquitectura LeNet 1	47
3.101 Estructura de la Red Neuronal Convolutacional LeNet 1 con Variación	47
3.102 Hiperparámetros de la Red Neuronal Convolutacional LeNet 1 con Variación	47
3.103 Matriz de Confusión Arquitectura LeNet 1.2	47
3.104 Porcentaje de Error y exactitud Arquitectura LeNet 1 con Variación	48
3.105 Métricas Arquitectura LeNet 1 con Variación	48
3.106 Estructura de la Red Neuronal Convolutacional LeNet 5	48
3.107 Hiperparámetros de la Red Neuronal Convolutacional LeNet 1 con Variación	49
3.108 Matriz de Confusión Arquitectura LeNet 5	49
3.109 Porcentaje de Error y exactitud Arquitectura LeNet 5	49
3.110 Métricas Arquitectura LeNet 5	49
3.111 Estructura de la Red Neuronal Convolutacional LeNet 5 con Variación	50
3.112 Hiperparámetros de la Red Neuronal Convolutacional LeNet 1 con Variación	50
3.113 Matriz de Confusión Arquitectura LeNet 5.2	50
3.114 Porcentaje de Error y exactitud Arquitectura LeNet 5.2	50
3.115 Métricas Arquitectura LeNet 5 con Variación	51
3.116 Estructura de la Red Neuronal Convolutacional LeNet 1	51
3.117 Hiperparámetros de la Red Neuronal Convolutacional LeNet 1	51
3.118 Matriz de Confusión Arquitectura LeNet 1	52
3.119 Porcentaje de Error y exactitud Arquitectura LeNet 1	52
3.120 Métricas Arquitectura LeNet 1	52
3.121 Estructura de la Red Neuronal Convolutacional LeNet 1 con Variación	52
3.122 Hiperparámetros de la Red Neuronal Convolutacional LeNet 1 con Variación	53
3.123 Matriz de Confusión Arquitectura LeNet 1.2	53

3.124	Porcentaje de Error y exactitud Arquitectura LeNet 1 con Variación	53
3.125	Métricas Arquitectura LeNet 1 con Variación	53
3.126	Estructura de la Red Neuronal Convolutacional LeNet 5	54
3.127	Hiperparámetros de la Red Neuronal Convolutacional LeNet 1 con Variación	54
3.128	Matriz de Confusión Arquitectura LeNet 5	54
3.129	Porcentaje de Error y exactitud Arquitectura LeNet 5	54
3.130	Métricas Arquitectura LeNet 5	55
3.131	Estructura de la Red Neuronal Convolutacional LeNet 5 con Variación	55
3.132	Hiperparámetros de la Red Neuronal Convolutacional LeNet 1 con Variación	55
3.133	Matriz de Confusión Arquitectura LeNet 5.2	56
3.134	Porcentaje de Error y exactitud Arquitectura LeNet 5.2	56
3.135	Métricas Arquitectura LeNet 5 con Variación	56
3.136	Porcentaje de F1 Score Arquitectura LeNet 5 con Variación	57
3.137	Porcentaje de F1 Score Arquitectura LeNet 1	57

Resumen

Este trabajo presenta un modelo inteligente capaz de clasificar pacientes sanos, pacientes con neumonía y pacientes con covid-19 a través de imágenes de rayos x pulmonares, utilizando técnicas de procesamiento digital de imágenes e inteligencia artificial. Concretamente, la aplicación que se desarrolló está constituida por el preprocesamiento de las imágenes y la creación de una red neuronal convolucional, la cual dio como resultados 90.6923 % de F1 Score para la enfermedad de Covid-19, 92.1951 % de F1 Score para la enfermedad de Neumonía y 96.3871 % de F1 Score para los pacientes sanos, la cual es la métrica que definen el modelo como exacto, eficiente y con sensibilidad a los cambios en las pruebas que realiza.

Cabe resaltar que la red neuronal convolucional que arrojó dichos resultados fue la que comprende una arquitectura basada en: primera capa de convolución con 6 filtros, función de activación ReLu y kernel de inicialización Glorot Normal, capa de MaxPooling de 2x2, segunda capa de convolución con 16 filtros, función de activación ReLu y kernel de inicialización Glorot Normal, capa de MaxPooling de 2x2 y finaliza con una 3 capas densas totalmente conectada que contiene 182 neuronas y función de activación ReLu para la salida y cuenta con una función de activación SoftMax; de igual manera se presentan diferentes tipos de pruebas preliminares que se mostrarán en el capítulo 3, con el fin de comparar el desempeño final.

Abstract

This work presents an intelligent model capable of classifying healthy, Pneumonia, and covid-19 patients through lung X-Ray images using digital image processing techniques and artificial intelligence. Specifically, the application to be developed consists of the pre-processing of the images and the creation of a convolutional neural network, which resulted in 90.6923 % of the F1 Score for the Covid-19 disease, 92.1951 % of the F1 Score for the illness of Pneumonia and 96.3871 % of the F1 Score for healthy patients, which is the metric that defines the model as accurate, efficient, and sensitive to changes in the tests it performs.

It should be noted that the convolutional neural network that yielded these results was the one that includes an architecture based on: first convolution layer with six filters, ReLu activation function and Glorot Normal initialization kernel, 2x2 MaxPooling layer, second convolution layer with 16 filters, ReLu gating function and Glorot Normal initialization kernel, 2x2 MaxPooling layer and ending with a fully connected three dense layers containing 182 neurons and ReLu gating for output and featuring a SoftMax gating function; In the same way, different types of preliminary tests are presented that are shown in chapter 3, to compare the final performance.

CAPÍTULO 1

Introducción

1.1	Introducción	1
1.2	Antecedentes	2
1.3	Justificación	3
1.4	Objetivos	4
1.4.1	Objetivo general	4
1.4.2	Objetivos específicos	4
1.5	Planteamiento general del problema	4
1.6	Organización de la tesis	5

1.1. Introducción

La detección temprana de afecciones pulmonares es fundamental para aumentar la calidad de vida de las personas. Las enfermedades del tracto respiratorio son una de las causas más frecuentes e importantes de enfermedad humana, en EE.UU. se estima que anualmente se producen alrededor de 4 millones de casos de neumonía adquirida de los cuales 600 mil resultan en hospitalizaciones; además, la neumonía es la sexta causa principal de muerte y la primera causa de muerte por infección.

Según los datos de la Organización Mundial de la Salud (OMS), la exposición a la contaminación del aire causa 7 millones de muertes prematuras, en los niños, esto podría suponer una reducción del crecimiento y la función pulmonares, infecciones respiratorias y agravamiento del asma. [12]

Otra enfermedad infecciosa respiratoria es el coronavirus (COVID-19) la cual es causada por el virus SARS-CoV-2; enfermedad que para el año 2019 causó una pandemia a nivel mundial, dicha enfermedad tiene efecto mas notorio en personas mayores y que padezcan enfermedades subyacentes de tipo cardiovasculares, diabetes, enfermedades respiratorias crónicas o cáncer, éstas son aquellas que tienen más probabilidades de desarrollar una enfermedad grave que puede llegar a la muerte. [13]

Actualmente por medio de la técnica de resonancia magnética y tomografía computarizada se pueden llevar a cabo varios estudios científicos que traen consigo una mejor evaluación diagnóstica de pacientes haciendo de su tratamiento un trámite más ágil y eficiente.

Este trabajo tiene como objetivo principal, dar solución al problema de clasificación e identificación de enfermedades pulmonares por medio de imágenes biomédicas de Rayos X pulmonares, proponiendo un modelo computacional que sea de ayuda al personal médico al momento de realizar un diagnóstico temprano, ya que es de vital importancia la detección temprano de pacientes con posibles afecciones pulmonares.

1.2. Antecedentes

Hacia el año 1958, Frank Rosenblatt, se inspiró en el trabajo realizado por Warren McCulloch y Walter Pitts que dio como finalidad la creación del perceptrón, dando a su vez el inicio a las redes neuronales de una manera un tanto diferente a como las conocemos hoy en día. En Figura 1.1, se muestran los primeros esquemas para conceptualizar la organización de un perceptrón.

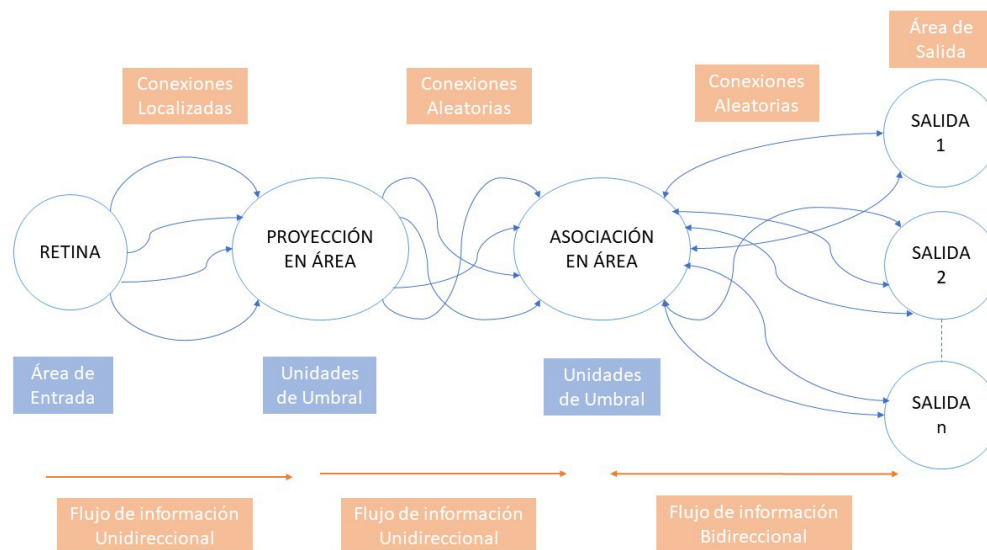


Figura 1.1. Organización de un perceptrón por Frank Rosenblatt.

Fuente:[30] Modificación: Autor

A medida que avanzan los años se van presentando nuevas mejoras tales como el perceptrón multicapa, el aprendizaje automático, las redes tipo feedforward, la técnica de backpropagation, las redes neuronales convolucionales, el deep learning, entre otras.

Muchos de los algoritmos propuestos en los años ochenta se tienen en consideración para dar soluciones a problemáticas de la actualidad, debido a esto se puede decir que las variantes presentadas a lo largo de los años han sido aportes de gran impacto ya que se puede llevar a cabo en diferentes ámbitos, con diferentes parámetros de entrada y obteniendo resultados aceptables en todas las áreas de estudio, no sólo en el ámbito de la computación, sino hoy en día son de gran ayuda en el ámbito médico.

En Figura 1.2, se muestran los primeros estudios para la obtención de imágenes médicas partir de rayos X y poder detectar padecimientos en pacientes.

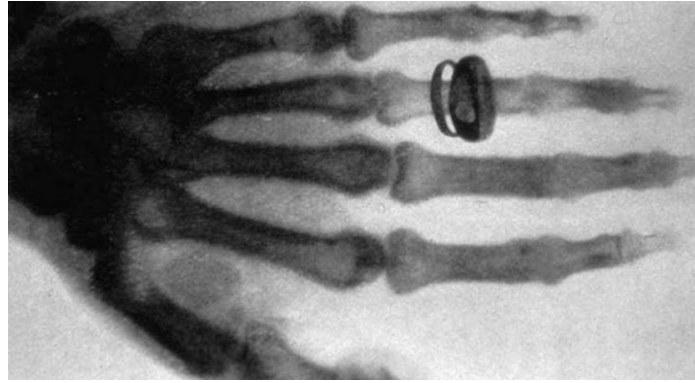


Figura 1.2. La primera radiografía de la historia: la mano de Anna Bertha Ludwig.
Fuente:[8].

A la par con los avances obtenidos en el área de la inteligencia artificial, se ven los avances en el área del procesamiento digital de imágenes; dos áreas que juntas pueden llegar a dar resultados de gran impacto a la sociedad. Para el desarrollo de este proyecto el caso de estudio va a ser problemática de salud pública.

En la actualidad se pueden encontrar diversas investigaciones que relacionan el procesamiento de imágenes biomédicas [16] y las redes neuronales [17], dando como resultado técnicas abordadas, modelos propuestos y demás recopilaciones de diferentes imágenes y sus respectivos resultados.

1.3. Justificación

En el área de la inteligencia artificial existen diferentes modelos predictivos los cuales se pueden clasificar: por la topología (monocapa o perceptrón simple y multicapa o perceptrón multicapa), por el tipo de conexión (no recurrentes y recurrentes) y por el grado de conexiones (Redes neuronales totalmente conectadas y Redes neuronales parcialmente conectadas).

Para el desarrollo de este proyecto se tomó como base fundamental la ventaja del procesamiento de imágenes y patrones visuales que tienen las redes neuronales convolucionales frente a las redes tipo perceptrón multicapa; ya que estas últimas son utilizadas para tareas más genéricas. En Figura 1.3, se muestra una arquitectura clásica para red neuronal convolucional (CNN).

La velocidad de entrenamiento de las redes neuronales convolucionales, es significativamente mayor en comparación con el resto de redes neuronales de la gran gama que existen hoy en día, por esto el hecho de dicho atributo fuese en gran medida un factor influyente para la implementación en este proyecto.

Adicionalmente, gracias al uso de funciones de activación modernas para redes convolucionales, se reduce el número de conexiones y de parámetros, y aumenta la estabilidad en la etapa de entrenamiento ya que no permite caer en la linealidad en el análisis de los datos. Algunos de los cálculos que se utilizan en este tipo de redes neuronales son: Sigmoide, ReLu, SoftMax y Leaky ReLu [7]

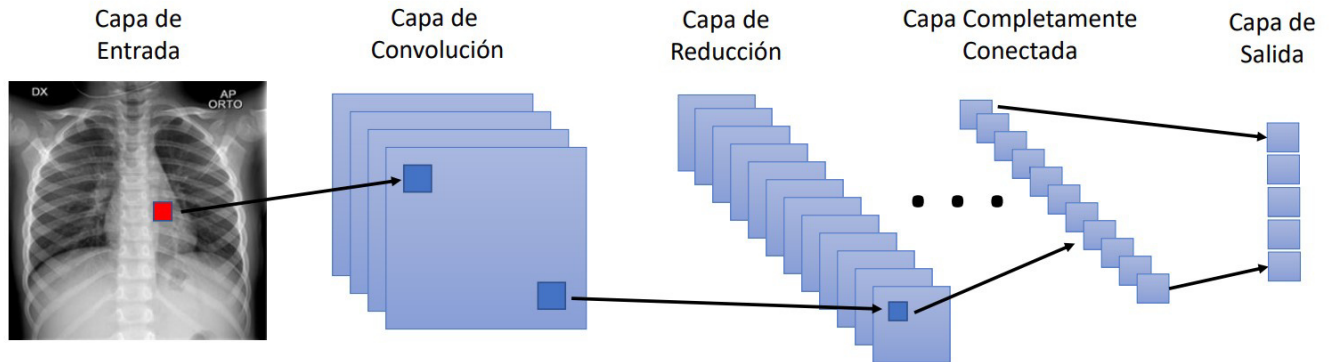


Figura 1.3. Estructura de una red neuronal convolucional (CNN)
Fuente: [15]. Modificación: Autor

1.4. Objetivos

1.4.1. Objetivo general

Diseñar un modelo computacional aplicando técnicas de inteligencia artificial y procesamiento digital de imágenes para la detección de pacientes con neumonía, covid-19 y pacientes sanos mediante imágenes biomédicas torácicas.

1.4.2. Objetivos específicos

1. Filtrar los atributos recolectados de la base de datos con el fin de seleccionar aquellos que serían los principales para dar un dictamen concreto.
2. Establecer técnicas de procesamiento digital de imágenes para detallar los atributos principales con los cuales se debe trabajar.
3. Diseñar una red neuronal convolucional con el fin de leer la base de datos existente y detectar algún tipo de enfermedad por medio del análisis radiográfico.
4. Realizar el entrenamiento de la red neuronal convolucional tomando como entrada radiografías torácicas para la detección de enfermedades pulmonares.
5. Validar los resultados obtenidos y comprobar la efectividad del modelo realizado, dando como resultados tablas comparativas de diferentes métricas de la propuesta dada en este proyecto.

1.5. Planteamiento general del problema

El procesamiento de imágenes hoy en día juegan un papel muy importante en áreas de investigación como lo son en la medicina, más específicamente en la detección de anomalías a nivel médico; de igual manera los avances en la tecnología arrojan resultados cada vez mejores en cuanto a resolución de las imágenes como de sus espectros.

En la actualidad, la inteligencia artificial realizada mediante redes neuronales convolucionales prometen ser de gran ayuda a la hora de realizar análisis, procesamiento y clasificación de imágenes; uno de sus mayores beneficios es la utilidad para reducir el tiempo de evaluación y diagnóstico de los pacientes.

Para este proyecto se propuso el diseño, construcción y validación de un modelo de detección de neumonía y covid-19 en imágenes de rayos X pulmonares, con el fin de obtener un programa computacional capaz de clasificar pacientes con neumonía, pacientes con covid-19 y pacientes sanos.

1.6. Organización de la tesis

Este trabajo de tesis está organizado en 5 capítulos, los cuales se describen brevemente a continuación:

Capítulo 1: Presenta la introducción y los objetivos de la presente obra.

Capítulo 2: Describe los conceptos necesarios para la creación de una red neuronal convolucional.

Capítulo 3: Presenta algoritmos comparativos para medir la eficiencia del modelo propuesto final.

Capítulo 4: Presenta algoritmo final implementado con sus parámetros de uso.

2.1	Conceptos de Redes Neuronales	7
2.2	Funciones de Activación	7
2.2.1	Función Sigmoide	7
2.2.2	Función ReLu (Rectificador Linear Unitario)	8
2.2.3	Función Leaky ReLu	8
2.2.4	Función SoftMax	9
2.3	Aprendizaje	9
2.3.1	Procesamiento de las imágenes	9
2.3.2	Convolución	11
2.3.3	Reducción/Pooling	11
2.3.4	Normalización del Batch (BatchNormalization)	12
2.3.5	Dropout	13
2.4	Función de pérdidas o coste	13
2.4.1	Error medio cuadrático	14
2.4.2	Error medio absoluto	14
2.4.3	Entropía cruzada categórica (Categorical Cross-Entropy)	14
2.4.4	Entropía cruzada binaria (Binary Cross-Entropy)	14
2.5	Optimizadores de función de pérdidas o coste	15
2.5.1	Descenso estocástico del gradiente (SGD)	15
2.5.2	Método adaptativo (Adagrad)	16
2.5.3	RMSprop	16
2.5.4	Estimación Adaptativa de Momentos (Adam)	17
2.5.5	Estimación del Momento Adaptativo acelerada por Nesterov (Nadam)	17
2.6	Kernel de Inicialización	18
2.6.1	GLOROT	18
2.6.2	HE	18
2.7	Métricas de Validación	19
2.7.1	Matriz de Confusión	19

2.7.2	Precisión	19
2.7.3	Sensibilidad (Recall)	20
2.7.4	Especificidad	20
2.7.5	Exactitud (Accuracy)	20
2.7.6	F1 Score	20

2.1. Conceptos de Redes Neuronales

Las redes neuronales artificiales, inicialmente llamadas Perceptrón Multicapa (PM) son una herramienta buena para la solución de problemas de clasificación. Se ha demostrado que los PM con una capa oculta en su arquitectura pueden separar satisfactoriamente las clases involucradas en un problema dado; sin embargo el número de nodos ocultos es desconocido ya que no existe un método para definirlos y depende mucho del problema a resolver, [20]

Para poder empezar con los componentes de una red neuronal convolucional, se debe partir de la arquitectura básica diseñada para una red neuronal sencilla; la cual esta compuesta por: Neuronas, conexiones, capas y funciones de activación; cabe resaltar que dicha arquitectura lo que propone es emular el comportamiento del cerebro humano.

Según el Instituto Nacional del Cáncer de EE.UU define una neurona como un tipo de célula que recibe y envía mensajes entre el cuerpo y el encéfalo, estos mensajes se envían por medio de una corriente eléctrica débil [3].

Así mismo, la conexión entre las neuronas se denomina sinapsis, la cual se basa en impulsos nerviosos que transmiten sustancias químicas entre dos o mas neuronas llamados neurotransmisores [4].

2.2. Funciones de Activación

Para poder dar a entender la organización entre las neuronas, y sus conexiones, se debe proceder a explicar la organización general de dichas arquitecturas; para esto se le denominan capas a cada sección por la cual se realiza un arreglo y/o acondicionamiento a la estructura general, como se muestra en Figura 2.1.

Internamente en cada capa de la red neuronal es necesario tener una función de activación, la cual gobierna el funcionamiento de las neuronas que intervienen en dicha etapa, es decir, devuelven una salida a partir de un valor de entrada. A lo largo de los años han ido evolucionando este tipo de funciones; las que mejores resultados han arrojado en diversos problemas propuestos son: Función Sigmoide, ReLu, Leaky ReLu y SoftMax.

Cada función de activación esta dada por una ecuación característica como se muestra a continuación:

2.2.1. Función Sigmoide

La función Sigmoide transforma los valores introducidos a una escala de 0 a 1, donde los valores altos tienen de manera asintótica a 1 y los valores muy bajos tienden de manera asintótica a 0 [2].

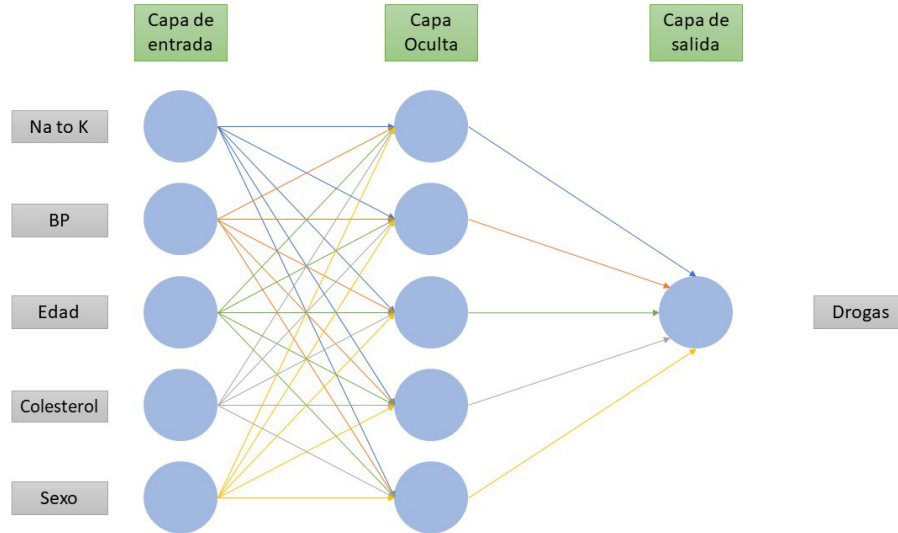


Figura 2.1. Estructura de una red neuronal.
Fuente: [21] Modificación: Autor

$$f(x) = \frac{1}{1 - e^x} \quad (2.1)$$

Esta función expresada en la ecuación 2.1 se caracteriza porque satura y mata el gradiente, tiene lenta convergencia, no tiene su centro en cero, esta acotada entre 0 - 1 y tiene buen rendimiento en la última capa.

2.2.2. Función ReLu (Rectificador Linear Unitario)

La función ReLu transforma los valores introducidos anulando los valores negativos y dejando los positivos tal y como ingresan [2].

$$\left. \begin{array}{l} 0 = x < 0 \\ x = x \geq 0 \end{array} \right\} \quad (2.2)$$

Esta función expresada en la ecuación 2.2 se caracteriza porque se activa sólo ante entradas positivas, no tiene acotación, tiene buen comportamiento con imágenes como entrada, se pueden morir neuronas en su utilización y tiene buen desempeño con redes neuronales convolucionales

2.2.3. Función Leaky ReLu

La función Leaky ReLu transforma los valores introducidos multiplicando los negativos por un coeficiente de rectificación y dejando los positivos según como entran [2].

$$\left. \begin{array}{l} 0 = x < 0 \\ a \cdot x = x \geq 0 \end{array} \right\} \quad (2.3)$$

Esta función expresada en la ecuación 2.3 se caracteriza porque penaliza los valores negativos mediante un coeficiente rectificador, no tiene acotamiento, tiene buen comportamiento cuando tiene imágenes como valores de entrada y buen desempeño con redes neuronales convolucionales.

2.2.4. Función SoftMax

La función SoftMax transforma las salidas a una representación en forma de probabilidades, de tal manera que la sumatoria de todas las probabilidades de salida sea 1. [2].

$$f(z)_j = \frac{e^{z_j}}{\sum_{k=1}^k e^{z_k}} \quad (2.4)$$

Esta función expresada en la ecuación 2.4 se caracteriza porque se utiliza para normalizar salidas multiclase, esta acotada entre 0-1, se obtiene una representación en forma de probabilidades según cada clase de salida que se tenga y además tiene buen rendimiento en la última capa.

2.3. Aprendizaje

El aprendizaje es el proceso por el cual una red neuronal modifica sus pesos en respuesta a una información de entrada. Los cambios que se producen durante el proceso de aprendizaje son la destrucción, modificación y creación de conexiones entre las neuronas.

Existe el aprendizaje supervisado, donde se parte de un conjunto de datos previamente etiquetados, es decir, se conoce el atributo objetivo de salida para el conjunto de datos que se dispone; se conoce también el aprendizaje No supervisado, el cual no contiene datos previamente etiquetados y con esto se procede a realizar un entrenamiento de forma autónoma [28].

2.3.1. Procesamiento de las imágenes

Una imagen digital es una representación bidimensional de una imagen a partir de una matriz numérica, frecuentemente binaria. Estas imágenes, generalmente se obtienen al convertir señales continuas en formato digital, las cuales se pueden visualizar en diversos medios como impresoras digitales, monitores y dispositivos de proyección digital. [27]

El procesamiento digital de imágenes, comprende un conjunto de técnicas aplicadas a imágenes digitales, esto con el fin de mejorar su calidad o facilitar la búsqueda de información en ellas.

Dentro de las técnicas más conocidas en el procesamiento de imágenes para el reconocimiento de patrones se encuentran:

- Adquisición El proceso de adquisición hace referencia al proceso de convertir un documento a una representación apta para ser procesada por un computador; dicha adquisición y conversión se puede

realizar mediante un escáner, una cámara fotográfica, etc, y el resultado pudiera variar dependiendo del proceso de digitalización usado y el método de codificación implementado en el proceso

- Binarización El proceso de binarización de una imagen consiste en separar el fondo de una imagen de los objetos que se requieran analizar, para obtener al final una imagen binaria, es decir que únicamente sus píxeles pueden tomar el valor de 0 ó 1, así como se puede ver en la Figura 2.2.

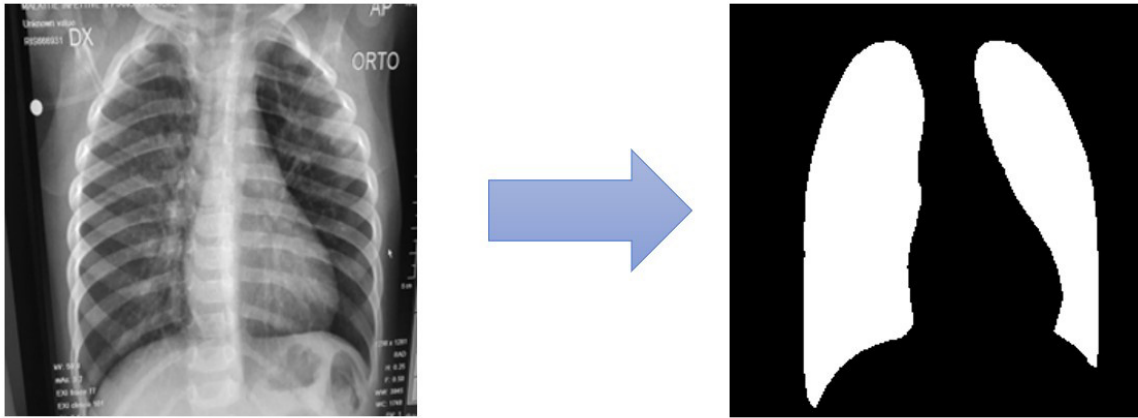


Figura 2.2. Proceso de Binarización de una imagen de rayos X
Fuente: Autor.

- Segmentación El proceso de segmentación consiste en delimitar las regiones de interés; para el caso de reconocer caracteres se refiere a líneas, bordes y caracteres individuales, así como se puede ver en la Figura 2.3.

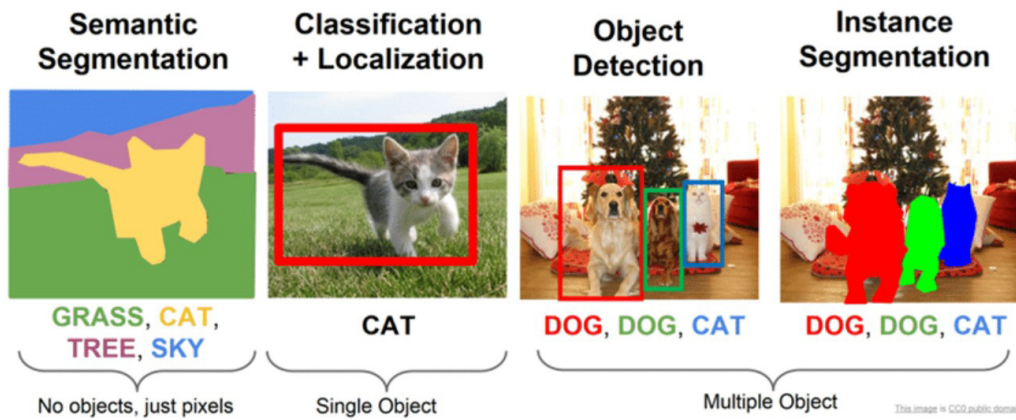


Figura 2.3. Diferentes tipos de Segmentación
Fuente: [26].

- Normalización El proceso de normalización hace referencia a ajustar el tamaño, la forma y la posición de los objetos presentes dentro de la imagen, con el fin de minimizar las variaciones entre las imágenes provenientes de la misma clase, así como se puede ver en la Figura 2.4.
- Extracción de Características El proceso de extracción de características permite conocer e identificar las características propias de la imagen en cuestión, éstas pueden ser área, perímetro, tamaño, color, textura, variación estándar, entre otras, así como se puede ver en la Figura 2.5 [24].

1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	0	0	0	0	1	1	1
1	1	1	0	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	1
1	1	1	0	0	0	0	1	1	1
1	1	1	0	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	1
1	1	1	0	0	0	0	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

Figura 2.4. Ejemplo de normalización de una imagen.
Fuente: [9].

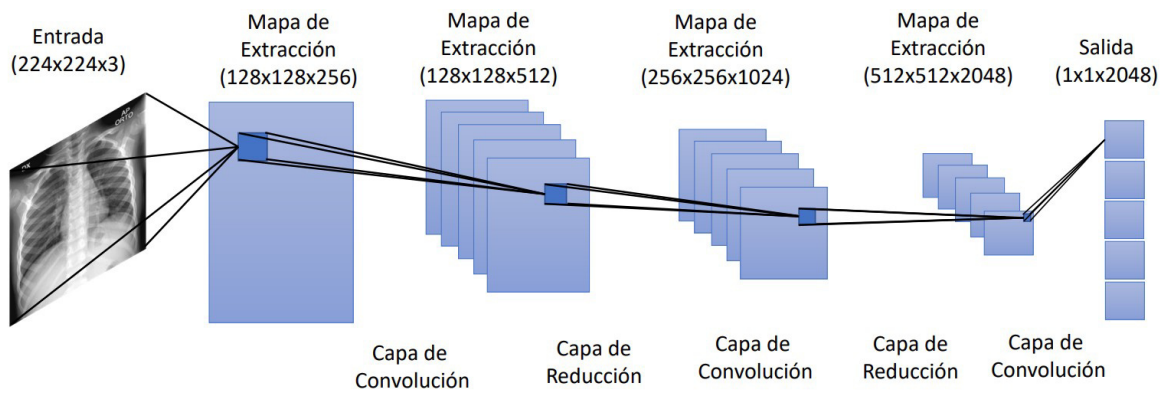


Figura 2.5. Diagrama de extracción de características mediante una CNN
Fuente: [22]. Modificación: Autor

2.3.2. Convolución

La capa convolucional está conformada por un conjunto de filtros (Kernels) que son convolucionados con una determinada entrada obteniendo en la salida un mapa de características; una vez se realiza el mapeado se aplica una función de activación con el fin de incrementar la no-linealidad proveniente de la imagen, así como se puede ver en la Figura 2.6.

2.3.3. Reducción/Pooling

La función principal de la capa de reducción o pooling es reducir el tamaño de la imagen de salida proveniente de la capa de convolución, esto con el fin de disminuir coste computacional, así como se puede ver en la Figura 2.7.

Las técnicas más usadas para esta capa son:

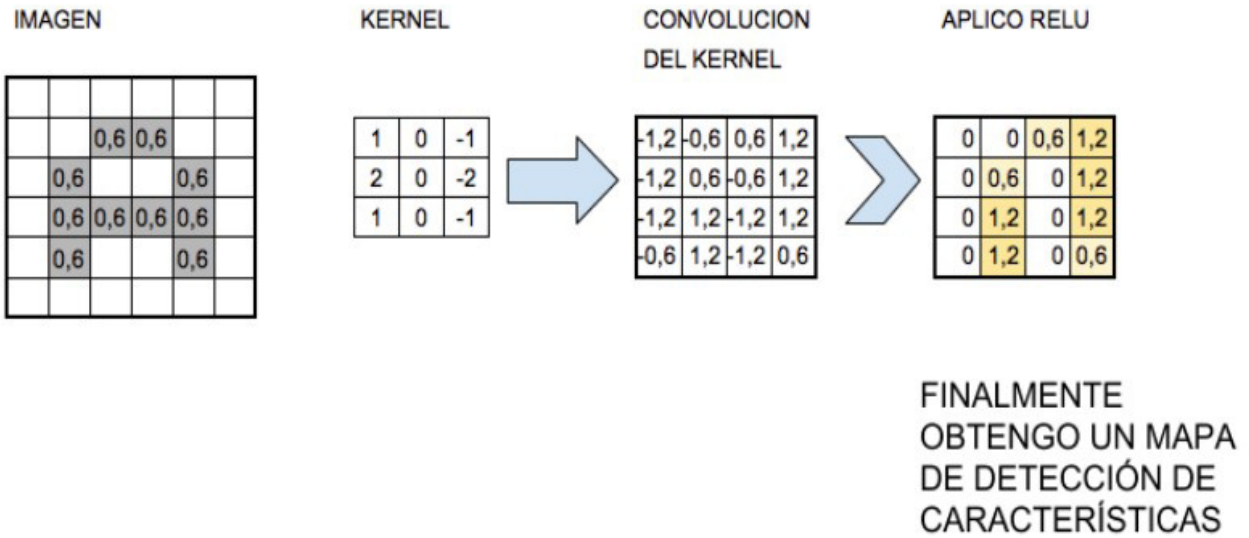


Figura 2.6. Ejemplo de Convolución con función de activación ReLu
Fuente: [1].

- Max-Pooling Ésta técnica consiste en dividir la imagen en regiones de las cuales se extrae el píxel de mayor valor [1].
- Average-Pooling Ésta técnica consiste en dividir la imagen en regiones de las cuales se extrae la media del valor de sus píxeles y devuelve el valor medio [1].

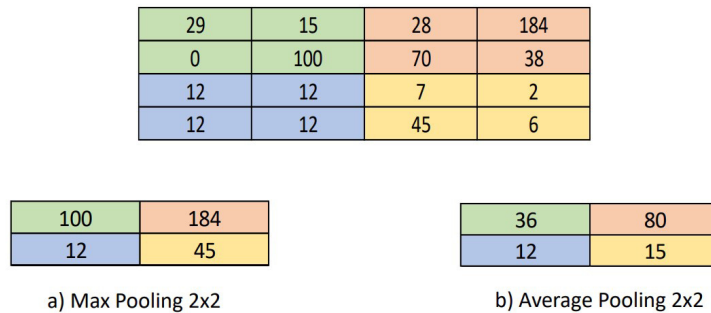


Figura 2.7. Ejemplo de los métodos de Pooling
Fuente: [23]. Modificado: Autor.

2.3.4. Normalización del Batch (BatchNormalization)

La normalización en lotes consiste en añadir un paso extra entre las neuronas y la función de activación para así normalizar las activaciones de salida. La normalización se hace usando la media y la varianza de todo el conjunto de entrenamiento excepto en el caso de que se emplee el descenso de gradiente estocástico que se usará a la media y la varianza de cada mini-lote de entrada. La normalización por lotes es una técnica de ayuda al entrenamiento que ayuda a que la convergencia hacia el mínimo global se produzca más rápidamente.

Aplicando el parámetro momentum cuando se introduce un nuevo mini-batch de entrada se usan una media y una desviación similares a las de la iteración anterior. Gracias a esto se consigue reducir el sobre-ajuste [6].

2.3.5. Dropout

El método de Dropout también llamada técnica de regularización, se basa en la eliminación de neuronas de una red neuronal de forma aleatoria, con el fin de disminuir el sobre-entrenamiento (overfitting), ya que obliga a las neuronas cercanas a las eliminadas de alguna manera a no aprender patrones de relación entre estas así como se puede ver en la Figura 2.7.

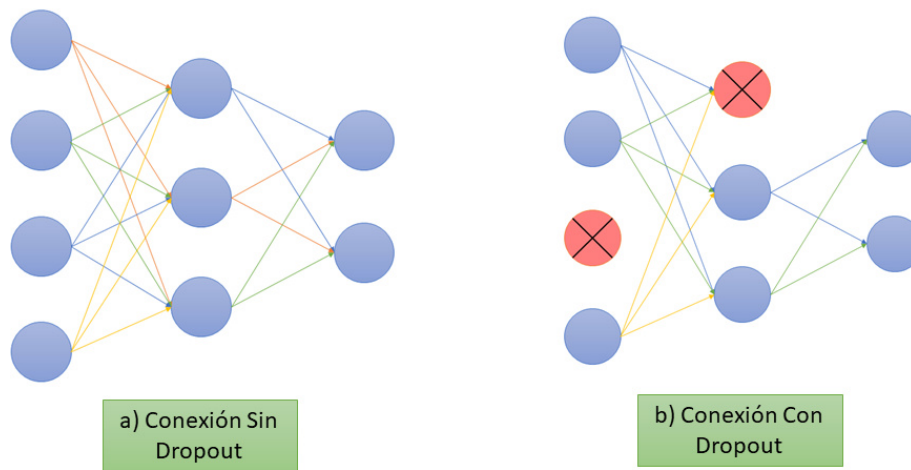


Figura 2.8. Red antes y después de aplicar Dropout
Fuente: [10]. Modificado: Autor.

Los valores que puede tomar este parámetro se encuentran entre 0 y 1; adicionalmente se puede decir que éste parámetro se utiliza únicamente en el periodo de entrenamiento.

2.4. Función de pérdidas o coste

Esta función es empleada para determinar el error entre el valor estimado y el valor real, esto con el fin de optimizar y determinar los parámetros de la red neuronal; dentro de las funciones mas comunes utilizadas se encuentran las siguientes:

2.4.1. Error medio cuadrático

El error medio cuadrático es una medida de precisión calculada como la raíz cuadrada media de los residuos. Se entiende como residuos la diferencia entre el valor real y el valor obtenido [2].

$$RMSE = \sqrt{\sum_{i=1}^n \frac{(\hat{y}_i - y_i)^2}{n}} \quad (2.5)$$

Esta función expresada en la ecuación 2.5 se caracteriza por penalizar los valores que son muy grandes y funciona muy bien para optimizar regresiones en general.

2.4.2. Error medio absoluto

El error medio absoluto es una medida de precisión y se calcula como al suma media de los valores absolutos de los errores [2].

$$MAE = \sum_{i=1}^n \frac{|\hat{y}_i - y_i|}{n} \quad (2.6)$$

Esta función expresada en la ecuación 2.6 se caracteriza por penalizar menos los valores grandes, es de fácil interpretación y es más difícil su diferenciación y convergencia.

2.4.3. Entropía cruzada categórica (Categorical Cross-Entropy)

La entropía cruzada categórica es una medida de precisión para variables categóricas (variable con un número limitado de valores distintos) [2].

$$\mathbb{L}(\Theta) = -\frac{1}{n} \sum_{i=1}^n \sum_{j=1}^m y_{ij} \log(p_{ij}) \quad (2.7)$$

Esta función expresada en la ecuación 2.7 se caracteriza por tener una escala univariable, es simétrica, de fácil interpretación y es más difícil su diferenciación y convergencia.

2.4.4. Entropía cruzada binaria (Binary Cross-Entropy)

La entropía cruzada binaria es una medida de precisión para variables binarias (variable que solo pueden tomar dos posibles valores) [2].

$$\mathbb{L}(\Theta) = -\frac{1}{n} \sum_{i=1}^n y_i \log(p_i + y) \log(1 - \Pi) \quad (2.8)$$

Esta función expresada en la ecuación 2.8 se caracteriza por tener una escala univariable, es simétrica, de fácil interpretación y es más difícil su diferenciación y convergencia.

2.5. Optimizadores de función de pérdidas o coste

2.5.1. Descenso estocástico del gradiente (SGD)

El descenso estocástico o gradiente descendiente es una aproximación estocástica del gradiente descendiente usado para minimizar una función objetivo que se escribe como una suma de funciones diferenciables [2].

En muchas de las técnicas de aprendizaje automático, el aprendizaje consiste en encontrar qué parámetros W minimizan la función de coste. Esto es así para la regresión lineal y polinómica, la regresión logística, el deep learning, etc. El gradiente descendiente es un método de optimización numérica para estimar los mejores coeficientes [18].

Por ejemplo: Se tiene una función f que es una línea pero no se sabe cuáles son sus parámetros, en ese caso se observa la ecuación 2.9

$$y = f(x) = wx + b \quad (2.9)$$

Para hacer mas simple este caso, el valor de b será 0. Una vez se aplica el gradiente descendiente por tanteo se obtiene que para $W=1$ se obtiene un MSE de 758.37, al aumentar o disminuir el valor de W se ve reflejado en el cambio de valor del MSE; para $W=2$, el MSE sería 265.82, para $W=3$, el MSE sería 30.02, para $W=4$, el MSE sería 50.97, para $W=3.5$, el MSE sería 8.40.

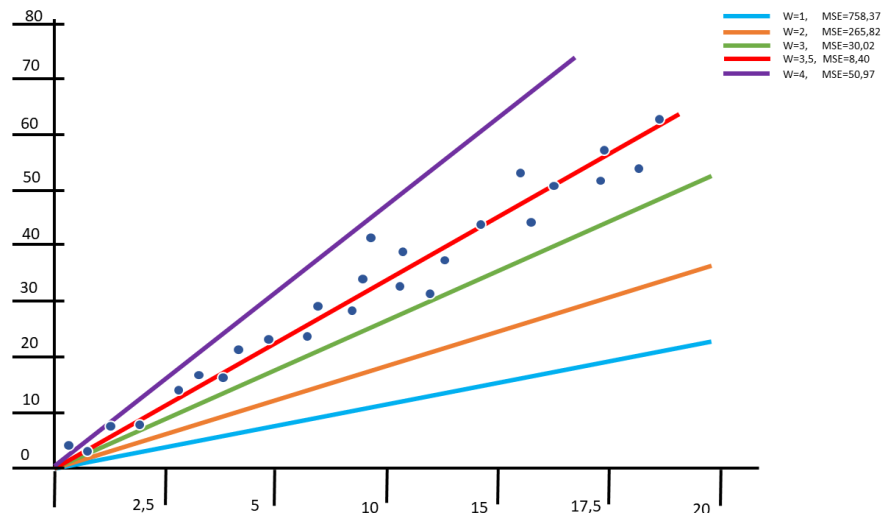


Figura 2.9. Ejemplo de SGD
Fuente: [18]. Modificado: Autor.

En términos matemáticos, para poder poner en uso la técnica del gradiente descendiente es necesario utilizar derivadas parciales del MSE en función del peso W como se expresa a continuación en la ecuación 2.10:

$$\frac{\partial MSE}{\partial W} \quad (2.10)$$

Una vez se tenga el valor de esta derivada parcial, se procede a actualizar los coeficientes utilizando las ecuaciones 2.11 y 2.12 :

$$W_{k+1} = W_k - \alpha \frac{\partial MSE}{\partial W_k} \quad (2.11)$$

$$b_{k+1} = b_k - \alpha \frac{\partial MSE}{\partial b_k} \quad (2.12)$$

Es necesario utilizar la resta, para ir en la dirección opuesta al gradiente, además la actualización del peso W es proporcional al gradiente, es decir, a mayor pendiente, mayor es el cambio del valor de W , adicionalmente se tiene un factor de aprendizaje involucrado (α) el cual tiene como función controlar el tamaño de la actualización.

2.5.2. Método adaptativo (Adagrad)

Este optimizador ajusta el factor de aprendizaje (learning rate) a los parámetros de la red, empleando las ecuaciones 2.13 y 2.14, donde actualizaciones más grandes se llevan a cabo sobre parámetros que infrecuentemente cambian, mientras que los que varían más seguidamente reciben un ajuste más pequeño [19].

$$C_{i+1} = C_i + \Delta W_i^2 \quad (2.13)$$

$$W_{i+1} = W_i - \frac{\alpha * \Delta W_i}{\sqrt{C_i + \epsilon}} \quad (2.14)$$

El parámetro C es un “cache” que mantiene la suma de los cuadrados de los gradientes (derivadas), por cada parámetro.

El parámetro ϵ es una pequeña constante para evitar la división por cero.

2.5.3. RMSprop

Este optimizador busca corregir los efectos negativos de la acumulación global de los caches, al convertirlo en un promedio móvil exponencial con pesos, con las ecuaciones 2.15 y 2.16 se representa la formulación del optimizador [19]:

$$C_i = C_i + d * C * (1 - d) * \Delta W_i^2 \quad (2.15)$$

$$W_{i+1} = W_i - \frac{\alpha * \Delta W_i}{\sqrt{C_i + \epsilon}} \quad (2.16)$$

El parámetro "**d**" es la tasa de descomposición/decadencia (decaying rate), es un hiperparámetro típicamente fijado en 0.9, el cual hace que las entradas anteriores en el cache influyan considerablemente menos que las nuevas.

El parámetro ϵ es un una pequeña constante para evitar la división por cero.

2.5.4. Estimación Adaptativa de Momentos (Adam)

El optimizador denominado Adam es una variación del optimizador RMSProp adicionando momentum, tal como y se muestra en las ecuaciones 2.17, 2.18 y 2.19:

$$m_i = \beta_1 * m_{i-1} + (1 - \beta_1) * \Delta W_i \quad (2.17)$$

$$v_i = \beta_2 * v_{i-1} + (1 - \beta_2) * \Delta W_i^2 \quad (2.18)$$

$$W_{i+1} = W_i - \frac{\alpha * m_i}{\sqrt{v_i} + \epsilon} \quad (2.19)$$

m y v representan los dos momentos, siendo m el que modela la media de los gradientes a lo largo del tiempo, mientras que v hace lo mismo con la varianza. β_1 es igual, en la mayoría de los casos, a 0.9, mientras que β_2 casi siempre se fija en 0.99 [19].

El parámetro ϵ es un una pequeña constante para evitar la división por cero.

2.5.5. Estimación del Momento Adaptativo acelerada por Nesterov (Nadam)

El optimizador denominado Nadam es una variación del optimizador RMSProp adicionando aceleración de Nesterov, tal como y se muestra en las ecuaciones 2.20, 2.21 y 2.22:

$$m_i = \beta_1 * m_{i-1} + (1 - \beta_1) * \Delta(W_i + \beta_1 * m_{i-1}) \quad (2.20)$$

$$v_i = \beta_2 * v_{i-1} + (1 - \beta_2) * \Delta(W_i + \beta_1 * m_{i-1})^2 \quad (2.21)$$

$$W_{i+1} = W_i - \frac{\alpha * m_i}{\sqrt{v_i} + \epsilon} \quad (2.22)$$

m y v representan los dos momentos, siendo m el que modela la media de los gradientes a lo largo del tiempo, mientras que v hace lo mismo con la varianza. β_1 es igual, en la mayoría de los casos, a 0.9, mientras que β_2 casi siempre se fija en 0.99 [19].

El parámetro ϵ es un una pequeña constante para evitar la división por cero.

2.6. Kernel de Inicialización

Los Kernel de inicialización dentro de una red neuronal convolucional hacen referencia a la asignación de valores iniciales a los pesos de conexión entre las neuronas. Dicha inicialización representa un entrenamiento efectivo para la red y por ello existen diferentes modelos propuestos; algunos de ellos son:

2.6.1. GLOROT

La idea principal propuesta por por Xavier Glorot es mencionada en [11] dónde se habla del factor de normalización al inicializar los pesos de las capas ocultas dela red que tenga en cuenta el tamaño de la capa de entrada y de salida.

- **Glorot Uniforme**

Esta técnica de inicialización de pesos se utiliza para mantener la varianza de las activaciones constantes a través de las capas. En este caso los pesos tienen una distribución uniforme y se muestra en la ecuación 2.23,

$$W_j \sim U \left(-\sqrt{\frac{6}{n_j + n_{j+1}}}, \sqrt{\frac{6}{n_j + n_{j+1}}} \right) \quad (2.23)$$

El parámetro \mathbf{W} hace referencia a los pesos, \mathbf{U} es la representación de la distribución uniforme y n_j, n_{j+1} son el número de capas de entrada y salida respectivamente.

- **Glorot Normal** Esta técnica de inicialización de pesos se utiliza para mantener la varianza de las activaciones constantes a través de las capas. En este caso los pesos tienen una distribución normal (Gaussiana) y se muestra en la ecuación 2.24,

$$W_j \sim N \left(0, \sqrt{\frac{2}{n_j + n_{j+1}}} \right) \quad (2.24)$$

El parámetro \mathbf{W} hace referencia a los pesos, \mathbf{N} es la representación de la distribución normal y n_j, n_{j+1} son el número de capas de entrada y salida respectivamente.

2.6.2. HE

La técnica propuesta por por Kaiming He es mencionada en [14] dónde se habla del factor de la inicialización de los pesos en redes neuronales profundas, especialmente aquellas que utilizan de activación ReLu.

- **He Uniforme**

$$W_j \sim U \left(-\sqrt{\frac{6}{n_j}}, \sqrt{\frac{6}{n_j}} \right) \quad (2.25)$$

El parámetro \mathbf{W} hace referencia a los pesos, \mathbf{U} es la representación de la distribución uniforme y n_j son el número de capas de entrada y salida respectivamente.

- **He Normal**

$$W_j \sim N \left(0, \sqrt{\frac{2}{n_j}} \right) \quad (2.26)$$

El parámetro \mathbf{W} hace referencia a los pesos, \mathbf{N} es la representación de la distribución normal y n_j son el número de capas de entrada y salida respectivamente.

2.7. Métricas de Validación

2.7.1. Matriz de Confusión

La matriz de confusión tiene como finalidad evaluar los resultados obtenidos a la salida de las pruebas; en este caso de una red neuronal convolucional. Esta se divide en falsos positivos, falsos negativos, verdaderos positivos y verdaderos negativos. Con estos valores se puede calcular las métricas de precisión y Exhaustividad(Recall). [5].

- Verdaderos positivos (TP): La salida predicha por la red neuronal coincide con la salida esperada según los datos reales (Persona que tiene covid19 y el modelo lo clasificó como covid19 (+))
- Falsos positivos (FP): La salida predicha por la red neuronal NO coincide con la salida esperada según los datos reales (Persona que no tiene covid19 y el modelo lo clasificó como covid19 (+)).
- Falsos negativos (FN): La salida predicha por la red neuronal NO coincide con la salida esperada según los datos reales (Persona que tiene covid19 y el modelo lo clasificó como covid19 (-)).
- Verdaderos negativos (TN): La salida predicha por la red neuronal coincide con la salida esperada según los datos reales (Persona que no tiene covid19 y el modelo lo clasificó como covid19 (-)).

Una mejor manera de entender los resultados obtenidos es plasmándolos en una tabla, tal como se muestra en la tabla 2.1.

Tabla 2.1: Matriz de Confusión

Fuente: [5]. Modificado: Autor.

	Predicción Clase A	Predicción Clase B	Predicción Clase C
Pertenencia Clase A	Verdadero Positivo (A)	Falso Positivo (A,B)	Falso Positivo (A,C)
Pertenencia Clase B	Falso Negativo (B,A)	Verdadero Positivo (B)	Falso Positivo (B,C)
Pertenencia Clase C	Falso Negativo (C,A)	Falso Negativo (C,B)	Verdadero Positivo (C)

2.7.2. Precisión

Empleando esta métrica se puede determinar la calidad del modelo empleado en tareas de clasificación; para el caso de este proyecto concierne a la cantidad de pacientes bien clasificados según su patología proporcionada por las placas de Rayos X.

La fórmula con la cual se obtiene un valor numérico de esta métrica se observa en la ecuación 2.27.

$$precisión = \frac{TP}{TP + FP} \quad (2.27)$$

2.7.3. Sensibilidad (Recall)

Empleando esta métrica se puede determinar la cantidad de muestras tanto casos positivos como negativos que el modelo empleado es capaz de identificar; La sensibilidad se representa como la fracción de verdaderos positivos.

La fórmula con la cual se obtiene un valor numérico de esta métrica se observa en la ecuación 2.27.

$$recall = \frac{TP}{TP + FN} \quad (2.28)$$

2.7.4. Especificidad

Empleando esta métrica se puede determinar la cantidad de muestras de los casos negativos que el modelo propuesto ha clasificado correctamente. Expresa cuan bien puede el modelo detectar esa clase.

La fórmula con la cual se obtiene un valor numérico de esta métrica se observa en la ecuación 2.29.

$$Especificidad = \frac{TN}{TP + FN} \quad (2.29)$$

2.7.5. Exactitud (Accuracy)

Empleando esta métrica se puede determinar el porcentaje de casos que el modelo ha acertado; el problema que conlleva utilizar esta métrica es que en ciertos casos hace parecer que un modelo malo parezca mejor de lo que realmente es.

La fórmula con la cual se obtiene un valor numérico de esta métrica se observa en la ecuación 2.30.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.30)$$

2.7.6. F1 Score

Esta es otra métrica muy empleada ya que resume la precisión y sensibilidad en una sola métrica. Por ello es de gran utilidad cuando la distribución de las clases es desigual, por ejemplo cuando el número de pacientes con una condición es del 15% y el otro es 85%, lo que en el campo de la salud es bastante común.

La fórmula con la cual se obtiene un valor numérico de esta métrica se observa en la ecuación 2.31.

$$F1Score = \frac{2 * Precisión * Sensibilidad}{Precisión + Sensibilidad} \quad (2.31)$$

Pruebas de Arquitecturas

3.1	Introducción	21
3.2	Arquitecturas Libres	22
3.2.1	Selección 70/30	22
3.2.2	Selección 80/20	28
3.2.3	Selección 95/5	34
3.3	Arquitecturas Conocidas	40
3.3.1	Selección 70/30	41
3.3.2	Selección 80/20	46
3.3.3	Selección 95/5	51
3.4	Filtro 1: Modificación de la función de Activación	56
3.5	Filtro 2: Modificación del optimizador	57
3.6	Filtro 3: Modificación Kernel de Inicialización	57

3.1. Introducción

En este capítulo se abordarán diferentes arquitecturas implementadas con variaciones internas en cuanto a la cantidad de capas, neuronas, optimizadores, funciones de activación y kernels inicializadores, así mismo se muestran todas las pruebas preliminares para poder llegar a un modelo más óptimo.

La base de datos [25] utilizada para realizar todos los experimentos fue tomada de un equipo de investigadores de la Universidad de Qatar, Doha, Qatar, y la Universidad de Dhaka, Bangladesh, junto con sus colaboradores de Pakistán y Malasia, en colaboración con médicos, han creado una base de datos de imágenes de radiografías de tórax para casos positivos de COVID-19 junto con Imágenes de Neumonía Normal y Viral. Este conjunto de datos sobre infecciones pulmonares normales, COVID-19 y otras infecciones se publica por etapas. En la actualización utilizada para el desarrollo de dichas pruebas se utilizaron 3.616 imágenes de COVID-19, 1.345 de Neumonía Viral y 10.193 de Neumonía Normal.

Las imágenes iniciales tienen diferentes tamaños, por esto se realizó un tratamiento de imágenes iniciales el cual consistió en estandarizar el tamaño, y adicionalmente se descartó una cuarta enfermedad llamada Opacidad Pulmonar debido a la capacidad de procesamiento que se tenía en su momento.

3.2. Arquitecturas Libres

Las arquitecturas libres hacen referencia a arquitecturas propuestas por un usuario cualquiera, en este caso propuestas por la autora, esto con el fin de comprobar la efectividad de diferentes arquitecturas propuestas, no solo basándose en modelos propuestos por otros, sino también de la propia autoría. Estas arquitecturas tienen en común la sencillez y simplicidad de sus capas.

Como primera instancia, se realizan pruebas con diferentes valores de separación de la base de datos; es decir, se proponen 3 diferentes métodos de separación, el primero con 70 % de los datos para entrenamiento y 30 % para validación, la segunda propuesta es de 80 % para entrenamiento y 20 % para validación, y la última propuesta es de 95 % para entrenamiento y 5 % para validación.

Partiendo de esto, a continuación se muestran los resultados obtenidos en esta primera prueba.

3.2.1. Selección 70/30

Propuesta 1

Tabla 3.1: Estructura de la Red Neuronal Convolutiva Libre 1.

	Nombre	Estructura	Valor	Función de Activación	Tamaño del Kernel
Capa 1	Entrada	256×256	16 filtros	ReLU	3×3
	Convolución 1	256×256			
Capa 2	Max Pooling 1	2×2			
Capa 3	Full Conected				
Capa 4	Salida	3×3		SoftMax	3×3

Una vez realizado el entrenamiento y la validación de esta arquitectura propuesta en conjunto con los hiperparámetros mostrados en la tabla 3.2, se obtienen los siguientes resultados:

Tabla 3.2: Hiperparámetros de la Red Neuronal Convolutiva Libre 1

Hiperparámetros	
Optimizador	ADAM
Función de pérdidas	CATEGORICAL CROSSENTROPY
Tasa de aprendizaje	0.001

- Matriz de confusión

Tabla 3.3: Matriz de Confusión Arquitectura Propuesta 1

	Covid-19	Neumonía	Normal
Covid-19	833	21	252
Neumonía	9	373	32
Normal	39	255	2.975

- Porcentaje de error y Exactitud

Tabla 3.4: Porcentaje de error y exactitud Arquitectura Propuesta 1

Arquitectura	Error mínimo Entrenamiento	Accuracy Máximo Entrenamiento	Error mínimo Prueba	Accuracy Máximo Prueba
Libre 1	0.0197 %	99.5864 %	0.2338 %	93.2221 %

- Porcentaje de Precisión, Sensibilidad y F1 Score

Tabla 3.5: Métricas Arquitectura Propuesta 1

	Covid-19	Neumonía	Normal
Precisión	94.5516 %	89.0214 %	91.2856 %
Sensibilidad	75.1645 %	90.0966 %	97.894 %
F1 Score	83.8449 %	89.5558 %	94.4744 %

Propuesta 2

Tabla 3.6: Estructura de la Red Neuronal Convolutiva Propuesta 2

	Nombre	Estructura	Valor	Función de Activación	Tamaño del Kernel
Capa 1	Entrada Convolución 1	256×256 256×256	16 filtros	LeakyReLU	3×3
Capa 2	Max Pooling 1	2×2			
Capa 3	Full Conected				
Capa 4	Salida	3×3		SoftMax	3×3

Una vez realizado el entrenamiento y la validación de esta arquitectura propuesta en conjunto con los hiperparámetros mostrados en la tabla 3.7, se obtienen los siguientes resultados:

Tabla 3.7: Hiperparámetros de la Red Neuronal Convolutiva Propuesta 2

Hiperparámetros	
Optimizador	NADAM
Función de pérdidas	CATEGORICAL CROSSENTROPY
Tasa de aprendizaje	0.001

- Matriz de confusión

Tabla 3.8: Matriz de Confusión Arquitectura Propuesta 2

	Covid-19	Neumonía	Normal
Covid-19	782	5	319
Neumonía	19	343	52
Normal	30	14	2.995

- Porcentaje de error y Exactitud

Tabla 3.9: Porcentaje de Error y exactitud Arquitectura Propuesta 2

Arquitectura	Error mínimo entrenamiento	Accuracy Máximo Entrenamiento	Error mínimo Prueba	Accuracy Máximo Prueba
Libre 2	0.1070 %	96.6071 %	0.2093 %	92.4983 %

- Porcentaje de Precisión, Sensibilidad y F1 Score

Tabla 3.10: Métricas Arquitectura Propuesta 2

	Covid-19	Neumonía	Normal
Precisión	94.1034 %	94.7513 %	88.9780 %
Sensibilidad	70.7052 %	82.5024 %	98.5521 %
F1 Score	80.7434 %	88.4020 %	93.5206 %

Propuesta 3**Tabla 3.11:** Estructura de la Red Neuronal Convolutacional Propuesta 3

	Nombre	Estructura	Valor	Función de Activación	Tamaño del Kernel
Capa 1	Entrada Convolución 1	256×256 256×256	16 filtros	ReLU	3×3
Capa 2	Max Pooling 1	2×2			
Capa 3	Dropout		0.2		
Capa 4	Full Conected				
Capa 5	BatchNormalization				
Capa 6	Densa		40 neuronas	ReLu	
Capa 7	Salida	3×3		SoftMax	3×3

Una vez realizado el entrenamiento y la validación de esta arquitectura propuesta en conjunto con los hiperparámetros mostrados en la tabla 3.12, se obtienen los siguientes resultados:

Tabla 3.12: Hiperparámetros de la Red Neuronal Convolutacional Propuesta 3

Hiperparámetros	
Optimizador	ADAM
Función de pérdidas	CATEGORICAL CROSSENTROPY
Tasa de aprendizaje	0.001

- Matriz de confusión

Tabla 3.13: Matriz de Confusión Arquitectura Propuesta 3

	Covid-19	Neumonía	Normal
Covid-19	991	10	105
Neumonía	89	287	38
Normal	309	13	2.717

- Porcentaje de Error y Exactitud

Tabla 3.14: Porcentaje de Error y Exactitud Arquitectura Propuesta 3

Arquitectura	Error mínimo entrenamiento	Accuracy Máximo Entrenamiento	Error mínimo Prueba	Accuracy Máximo Prueba
Libre 3	0.0082 %	99.8120 %	0.2080 %	93.9021 %

- Porcentaje de Precisión, Sensibilidad y F1 Score

Tabla 3.15: Métricas Arquitectura Propuesta 3

	Covid-19	Neumonía	Normal
Precisión	71.3462 %	92.5806 %	95.0 %
Sensibilidad	89.6021 %	69.3236 %	89.4044 %
F1 Score	79.4388 %	79.2817 %	92.1173 %

Propuesta 4**Tabla 3.16:** Estructura de la Red Neuronal Convolutiva Propuesta 4

	Nombre	Estructura	Valor	Función de Activación	Tamaño del Kernel
Capa 1	Entrada Convolución 1	256×256 256×256	16 filtros	Sigmoid	3×3
Capa 2	Max Pooling 1	2×2			
Capa 3	Dropout		0.3		
Capa 4	Full Conected				
Capa 5	BatchNormalization				
Capa 6	Densa		40 neuronas	Sigmoid	
Capa 7	Salida	3×3		SoftMax	3×3

Una vez realizado el entrenamiento y la validación de esta arquitectura propuesta en conjunto con los hiperparámetros mostrados en la tabla 3.17, se obtienen los siguientes resultados:

Tabla 3.17: Hiperparámetros de la Red Neuronal Convolutiva Propuesta 4

Hiperparámetros	
Optimizador	ADAM
Función de pérdidas	CATEGORICAL CROSSENTROPY
Tasa de aprendizaje	0.001

- Matriz de confusión

Tabla 3.18: Matriz de Confusión Arquitectura Propuesta 4

	Covid-19	Neumonía	Normal
Covid-19	708	19	379
Neumonía	16	365	33
Normal	179	59	2.801

- Porcentaje de Error y Exactitud

Tabla 3.19: Porcentaje de Error y exactitud Arquitectura Propuesta 4

Arquitectura	Error mínimo entrenamiento	Accuracy Máximo Entrenamiento	Error mínimo Prueba	Accuracy Máximo Prueba
Libre 4	0.3271 %	86.4379 %	0.3484 %	85.7863 %

- Porcentaje de Precisión, Sensibilidad y F1 Score

Tabla 3.20: Métricas Arquitectura Propuesta 4

	Covid-19	Neumonía	Normal
Precisión	78.4053 %	82.3927 %	87.1770 %
Sensibilidad	64.0144 %	88.1642 %	92.1684 %
F1 Score	70.4828 %	85.1808 %	89.6033 %

Propuesta 5

Tabla 3.21: Estructura de la Red Neuronal Convolutiva Propuesta 5

	Nombre	Estructura	Valor	Función de Activación	Tamaño del Kernel
Capa 1	Entrada Convolución 1	256×256 256×256	16 filtros	LeakyReLU	3×3
Capa 2	Max Pooling 1	2×2			
Capa 3	Dropout		0.4		
Capa 4	Full Conected				
Capa 5	BatchNormalization				
Capa 6	Densa		40 neuronas	LeakyReLU	
Capa 7	Salida	3×3		SoftMax	3×3

Una vez realizado el entrenamiento y la validación de esta arquitectura propuesta en conjunto con los hiperparámetros mostrados en la tabla 3.22, se obtienen los siguientes resultados:

Tabla 3.22: Hiperparámetros de la Red Neuronal Convolutiva Propuesta 5

Hiperparámetros	
Optimizador	RMSprop
Función de pérdidas	CATEGORICAL CROSSENTROPY
Tasa de aprendizaje	0.001

- Matriz de confusión

Tabla 3.23: Matriz de Confusión Arquitectura Propuesta 5

	Covid-19	Neumonía	Normal
Covid-19	892	10	204
Neumonía	28	348	38
Normal	63	23	2.953

- Porcentaje de Error y Exactitud

Tabla 3.24: Porcentaje de Error y exactitud Arquitectura Propuesta 5

Arquitectura	Error mínimo entrenamiento	Accuracy Máximo Entrenamiento	Error mínimo Prueba	Accuracy Máximo Prueba
Libre 5	0.1182 %	97.4060 %	0.6193 %	91.9719 %

- Porcentaje de Precisión, Sensibilidad y F1 Score

Tabla 3.25: Métricas Arquitectura Propuesta 5

	Covid-19	Neumonía	Normal
Precisión	90.7426 %	91.3385 %	92.4256 %
Sensibilidad	80.6509 %	84.0579 %	97.1701 %
F1 Score	85.3997 %	87.5471 %	94.7385 %

3.2.2. Selección 80/20

Propuesta 1

Tabla 3.26: Estructura de la Red Neuronal Convolutiva Libre 1.

	Nombre	Estructura	Valor	Función de Activación	Tamaño del Kernel
Capa 1	Entrada	256×256	16 filtros	ReLU	3×3
	Convulsión 1	256×256			
Capa 2	Max Pooling 1	2×2			
Capa 3	Full Conected				
Capa 4	Salida	3×3		SoftMax	3×3

Una vez realizado el entrenamiento y la validación de esta arquitectura propuesta en conjunto con los hiperparámetros mostrados en la tabla 3.27, se obtienen los siguientes resultados:

Tabla 3.27: Hiperparámetros de la Red Neuronal Convolutiva Libre 1

Hiperparámetros	
Optimizador	ADAM
Función de pérdidas	CATEGORICAL CROSSENTROPY
Tasa de aprendizaje	0.001

- Matriz de confusión

Tabla 3.28: Matriz de Confusión Arquitectura Propuesta 1

	Covid-19	Neumonía	Normal
Covid-19	631	8	114
Neumonía	5	246	20
Normal	67	20	1.928

- Porcentaje de error y Exactitud

Tabla 3.29: Porcentaje de error y exactitud Arquitectura Propuesta 1

Arquitectura	Error mínimo Entrenamiento	Accuracy Máximo Entrenamiento	Error mínimo Prueba	Accuracy Máximo Prueba
Libre 1	0.0230 %	99.3503 %	0.2 %	93.4517 %

- Porcentaje de Precisión, Sensibilidad y F1 Score

Tabla 3.30: Métricas Arquitectura Propuesta 1

	Covid-19	Neumonía	Normal
Precisión	89.7581 %	89.7810 %	93.5014 %
Sensibilidad	83.7981 %	90.7749 %	95.6823 %
F1 Score	86.6758 %	90.2752 %	94.5793 %

Propuesta 2

Tabla 3.31: Estructura de la Red Neuronal Convolutiva Propuesta 2

	Nombre	Estructura	Valor	Función de Activación	Tamaño del Kernel
Capa 1	Entrada	256×256	16 filtros	LeakyReLU	3×3
	Convolución 1	256×256			
Capa 2	Max Pooling 1	2×2			
Capa 3	Full Conected				
Capa 4	Salida	3×3		SoftMax	3×3

Una vez realizado el entrenamiento y la validación de esta arquitectura propuesta en conjunto con los hiperparámetros mostrados en la tabla 3.32, se obtienen los siguientes resultados:

Tabla 3.32: Hiperparámetros de la Red Neuronal Convolutiva Propuesta 2

Hiperparámetros	
Optimizador	NADAM
Función de pérdidas	CATEGORICAL CROSSENTROPY
Tasa de aprendizaje	0.001

- Matriz de confusión

Tabla 3.33: Matriz de Confusión Arquitectura Propuesta 2

	Covid-19	Neumonía	Normal
Covid-19	658	2	93
Neumonía	12	238	21
Normal	57	8	1.950

- Porcentaje de error y Exactitud

Tabla 3.34: Porcentaje de Error y exactitud Arquitectura Propuesta 2

Arquitectura	Error mínimo entrenamiento	Accuracy Máximo Entrenamiento	Error mínimo Prueba	Accuracy Máximo Prueba
Libre 2	0.0892 %	97.5328 %	0.1875 %	93.6492 %

- Porcentaje de Precisión, Sensibilidad y F1 Score

Tabla 3.35: Métricas Arquitectura Propuesta 2

	Covid-19	Neumonía	Normal
Precisión	90.5089 %	94.5.9677 %	94.4767 %
Sensibilidad	87.3837 %	87.8228 %	96.7741 %
F1 Score	88.9189 %	91.7148 %	95.6116 %

Propuesta 3

Tabla 3.36: Estructura de la Red Neuronal Convolutacional Propuesta 3

	Nombre	Estructura	Valor	Función de Activación	Tamaño del Kernel
Capa 1	Entrada Convolución 1	256×256 256×256	16 filtros	ReLU	3×3
Capa 2	Max Pooling 1	2×2			
Capa 3	Dropout		0.2		
Capa 4	Full Conected				
Capa 5	BatchNormalization				
Capa 6	Densa		40 neuronas	ReLu	
Capa 7	Salida	3×3		SoftMax	3×3

Una vez realizado el entrenamiento y la validación de esta arquitectura propuesta en conjunto con los hiperparámetros mostrados en la tabla 3.37, se obtienen los siguientes resultados:

Tabla 3.37: Hiperparámetros de la Red Neuronal Convolutacional Propuesta 3

Hiperparámetros	
Optimizador	ADAM
Función de pérdidas	CATEGORICAL CROSSENTROPY
Tasa de aprendizaje	0.001

- Matriz de confusión

Tabla 3.38: Matriz de Confusión Arquitectura Propuesta 3

	Covid-19	Neumonía	Normal
Covid-19	639	1	113
Neumonía	11	141	119
Normal	280	6	1.729

- Porcentaje de Error y Exactitud

Tabla 3.39: Porcentaje de Error y Exactitud Arquitectura Propuesta 3

Arquitectura	Error mínimo entrenamiento	Accuracy Máximo Entrenamiento	Error mínimo Prueba	Accuracy Máximo Prueba
Libre 3	0.0092 %	99.7944 %	0.1972 %	94.3402 %

- Porcentaje de Precisión, Sensibilidad y F1 Score

Tabla 3.40: Métricas Arquitectura Propuesta 3

	Covid-19	Neumonía	Normal
Precisión	68.7096 %	95.2702 %	88.1693 %
Sensibilidad	84.8605 %	52.0952 %	85.8064 %
F1 Score	75.9358 %	67.3031 %	86.9718 %

Propuesta 4**Tabla 3.41:** Estructura de la Red Neuronal Convolutiva Propuesta 4

	Nombre	Estructura	Valor	Función de Activación	Tamaño del Kernel
Capa 1	Entrada Convolución 1	256×256 256×256	16 filtros	Sigmoid	3×3
Capa 2	Max Pooling 1	2×2			
Capa 3	Dropout		0.3		
Capa 4	Full Conected				
Capa 5	BatchNormalization				
Capa 6	Densa		40 neuronas	Sigmoid	
Capa 7	Salida	3×3		SoftMax	3×3

Una vez realizado el entrenamiento y la validación de esta arquitectura propuesta en conjunto con los hiperparámetros mostrados en la tabla 3.42, se obtienen los siguientes resultados:

Tabla 3.42: Hiperparámetros de la Red Neuronal Convolutiva "Propuesta 4

Hiperparámetros	
Optimizador	ADAM
Función de pérdidas	CATEGORICAL CROSSENTROPY
Tasa de aprendizaje	0.001

- Matriz de confusión

Tabla 3.43: Matriz de Confusión Arquitectura Propuesta 4

	Covid-19	Neumonía	Normal
Covid-19	482	12	259
Neumonía	8	251	12
Normal	106	42	1.867

- Porcentaje de Error y Exactitud

Tabla 3.44: Porcentaje de Error y exactitud Arquitectura Propuesta 4

Arquitectura	Error mínimo entrenamiento	Accuracy Máximo Entrenamiento	Error mínimo Prueba	Accuracy Máximo Prueba
Libre 4	0.3352 %	85.8881 %	0.3424 %	85.5544 %

- Porcentaje de Precisión, Sensibilidad y F1 Score

Tabla 3.45: Métricas Arquitectura Propuesta 4

	Covid-19	Neumonía	Normal
Precisión	80.8724 %	82.2950 %	87.3246 %
Sensibilidad	64.0106 %	92.6199 %	92.5508 %
F1 Score	71.4603 %	87.1527 %	89.91.9 %

Propuesta 5

Tabla 3.46: Estructura de la Red Neuronal Convolutiva Propuesta 5

	Nombre	Estructura	Valor	Función de Activación	Tamaño del Kernel
Capa 1	Entrada Convolución 1	256×256 256×256	16 filtros	LeakyReLU	3×3
Capa 2	Max Pooling 1	2×2			
Capa 3	Dropout		0.4		
Capa 4	Full Conected				
Capa 5	BatchNormalization				
Capa 6	Densa		40 neuronas	LeakyReLU	
Capa 7	Salida	3×3		SoftMax	3×3

Una vez realizado el entrenamiento y la validación de esta arquitectura propuesta en conjunto con los hiperparámetros mostrados en la tabla 3.47, se obtienen los siguientes resultados:

Tabla 3.47: Hiperparámetros de la Red Neuronal Convolutiva Propuesta 5

Hiperparámetros	
Optimizador	RMSprop
Función de pérdidas	CATEGORICAL CROSSENTROPY
Tasa de aprendizaje	0.001

- Matriz de confusión

Tabla 3.48: Matriz de Confusión Arquitectura Propuesta 5

	Covid-19	Neumonía	Normal
Covid-19	539	1	213
Neumonía	25	135	111
Normal	33	1	1.981

- Porcentaje de Error y Exactitud

Tabla 3.49: Porcentaje de Error y exactitud Arquitectura Propuesta 5

Arquitectura	Error mínimo entrenamiento	Accuracy Máximo Entrenamiento	Error mínimo Prueba	Accuracy Máximo Prueba
Libre 5	0.1061 %	97.2460 %	0.5439 %	90.8851 %

- Porcentaje de Precisión, Sensibilidad y F1 Score

Tabla 3.50: Métricas Arquitectura Propuesta 5

	Covid-19	Neumonía	Normal
Precisión	90.2847 %	98.5401 %	85.9436 %
Sensibilidad	71.5803 %	49.8154 %	98.3126 %
F1 Score	79.5185 %	66.1747 %	91.7129 %

3.2.3. Selección 95/5

Propuesta 1

Tabla 3.51: Estructura de la Red Neuronal Convolutiva Propuesta 1.

	Nombre	Estructura	Valor	Función de Activación	Tamaño del Kernel
Capa 1	Entrada Convolución 1	256×256 256×256	16 filtros	ReLU	3×3
Capa 2	Max Pooling 1	2×2			
Capa 3	Full Conected				
Capa 4	Salida	3×3		SoftMax	3×3

Una vez realizado el entrenamiento y la validación de esta arquitectura propuesta en conjunto con los hiperparámetros mostrados en la tabla 3.52, se obtienen los siguientes resultados:

Tabla 3.52: Hiperparámetros de la Red Neuronal Convolutiva Propuesta 1

Hiperparámetros	
Optimizador	ADAM
Función de pérdidas	CATEGORICAL CROSSENTROPY
Tasa de aprendizaje	0.001

- Matriz de confusión

Tabla 3.53: Matriz de Confusión Arquitectura Propuesta 1

	Covid-19	Neumonía	Normal
Covid-19	173	1	12
Neumonía	4	66	5
Normal	28	9	461

- Porcentaje de error y Exactitud

Tabla 3.54: Porcentaje de error y exactitud Arquitectura Propuesta 1

Arquitectura	Error mínimo Entrenamiento	Accuracy Máximo Entrenamiento	Error mínimo Prueba	Accuracy Máximo Prueba
Libre 1	00.0241 %	99.2590 %	0.2032 %	93.2806 %

- Porcentaje de Precisión, Sensibilidad y F1 Score

Tabla 3.55: Métricas Arquitectura Propuesta 1

	Covid-19	Neumonía	Normal
Precisión	84.3902 %	86.8421 %	96.4435 %
Sensibilidad	93.0107 %	88 %	92.5703 %
F1 Score	88.4910 %	87.4172 %	94.46772 %

Propuesta 2

Tabla 3.56: Estructura de la Red Neuronal Convolutiva Propuesta 2

	Nombre	Estructura	Valor	Función de Activación	Tamaño del Kernel
Capa 1	Entrada Convolución 1	256×256 256×256	16 filtros	LeakyReLU	3×3
Capa 2	Max Pooling 1	2×2			
Capa 3	Full Conected				
Capa 4	Salida	3×3		SoftMax	3×3

Una vez realizado el entrenamiento y la validación de esta arquitectura propuesta en conjunto con los hiperparámetros mostrados en la tabla 3.57, se obtienen los siguientes resultados:

Tabla 3.57: Hiperparámetros de la Red Neuronal Convolutiva Propuesta 2

Hiperparámetros	
Optimizador	NADAM
Función de pérdidas	CATEGORICAL CROSSENTROPY
Tasa de aprendizaje	0.001

- Matriz de confusión

Tabla 3.58: Matriz de Confusión Arquitectura Propuesta 2

	Covid-19	Neumonía	Normal
Covid-19	164	31	19
Neumonía	2	68	5
Normal	9	6	483

- Porcentaje de error y Exactitud

Tabla 3.59: Porcentaje de Error y exactitud Arquitectura Propuesta 2

Arquitectura	Error mínimo entrenamiento	Accuracy Máximo Entrenamiento	Error mínimo Prueba	Accuracy Máximo Prueba
Libre 2	0.0614 %	98.2894 %	0.1959 %	94.2028 %

- Porcentaje de Precisión, Sensibilidad y F1 Score

Tabla 3.60: Métricas Arquitectura Propuesta 2

	Covid-19	Neumonía	Normal
Precisión	93.7142 %	88.3116 %	95.2662 %
Sensibilidad	88.1720 %	90.6666 %	96.9879 %
F1 Score	90.8587 %	89.4736 %	96.1194 %

Propuesta 3**Tabla 3.61:** Estructura de la Red Neuronal Convolutacional Propuesta 3

	Nombre	Estructura	Valor	Función de Activación	Tamaño del Kernel
Capa 1	Entrada Convolución 1	256×256 256×256	16 filtros	ReLU	3×3
Capa 2	Max Pooling 1	2×2			
Capa 3	Dropout		0.2		
Capa 4	Full Conected				
Capa 5	BatchNormalization				
Capa 6	Densa		40 neuronas	ReLu	
Capa 7	Salida	3×3		SoftMax	3×3

Una vez realizado el entrenamiento y la validación de esta arquitectura propuesta en conjunto con los hiperparámetros mostrados en la tabla 3.62, se obtienen los siguientes resultados:

Tabla 3.62: Hiperparámetros de la Red Neuronal Convolutacional Propuesta 3

Hiperparámetros	
Optimizador	ADAM
Función de pérdidas	CATEGORICAL CROSSENTROPY
Tasa de aprendizaje	0.001

- Matriz de confusión

Tabla 3.63: Matriz de Confusión Arquitectura Propuesta 3

	Covid-19	Neumonía	Normal
Covid-19	113	5	68
Neumonía	0	71	4
Normal	2	23	473

- Porcentaje de Error y Exactitud

Tabla 3.64: Porcentaje de Error y Exactitud Arquitectura Propuesta 3

Arquitectura	Error mínimo entrenamiento	Accuracy Máximo Entrenamiento	Error mínimo Prueba	Accuracy Máximo Prueba
Libre 3	0.0057 %	99.8130 %	0.2 %	95.7839 %

- Porcentaje de Precisión, Sensibilidad y F1 Score

Tabla 3.65: Métricas Arquitectura Propuesta 3

	Covid-19	Neumonía	Normal
Precisión	98.2608 %	71.7171 %	86.7889 %
Sensibilidad	60.7526 %	94.6666 %	94.9799 %
F1 Score	75.0830 %	81.6091 %	90.6999 %

Propuesta 4**Tabla 3.66:** Estructura de la Red Neuronal Convolutiva Propuesta 4

	Nombre	Estructura	Valor	Función de Activación	Tamaño del Kernel
Capa 1	Entrada Convolución 1	256×256 256×256	16 filtros	Sigmoid	3×3
Capa 2	Max Pooling 1	2×2			
Capa 3	Dropout		0.3		
Capa 4	Full Conected				
Capa 5	BatchNormalization				
Capa 6	Densa		40 neuronas	Sigmoid	
Capa 7	Salida	3×3		SoftMax	3×3

Una vez realizado el entrenamiento y la validación de esta arquitectura propuesta en conjunto con los hiperparámetros mostrados en la tabla 3.67, se obtienen los siguientes resultados:

Tabla 3.67: Hiperparámetros de la Red Neuronal Convolutiva Propuesta 4

Hiperparámetros	
Optimizador	ADAM
Función de pérdidas	CATEGORICAL CROSSENTROPY
Tasa de aprendizaje	0.001

- Matriz de confusión

Tabla 3.68: Matriz de Confusión Arquitectura Propuesta 4

	Covid-19	Neumonía	Normal
Covid-19	130	1	55
Neumonía	2	62	11
Normal	37	8	453

- Porcentaje de Error y Exactitud

Tabla 3.69: Porcentaje de Error y exactitud Arquitectura Propuesta 4

Arquitectura	Error mínimo entrenamiento	Accuracy Máximo Entrenamiento	Error mínimo Prueba	Accuracy Máximo Prueba
Libre 4	0.3369 %	86.1565 %	0.3429 %	86.1660 %

- Porcentaje de Precisión, Sensibilidad y F1 Score

Tabla 3.70: Métricas Arquitectura Propuesta 4

	Covid-19	Neumonía	Normal
Precisión	76.9230 %	87.3239 %	87.2832 %
Sensibilidad	69.8924 %	82.6666 %	90.9638 %
F1 Score	73.2394 %	84.9315 %	89.0855 %

Propuesta 5

Tabla 3.71: Estructura de la Red Neuronal Convolutiva Propuesta 5

	Nombre	Estructura	Valor	Función de Activación	Tamaño del Kernel
Capa 1	Entrada Convolución 1	256×256 256×256	16 filtros	LeakyReLU	3×3
Capa 2	Max Pooling 1	2×2			
Capa 3	Dropout		0.4		
Capa 4	Full Conected				
Capa 5	BatchNormalization				
Capa 6	Densa		40 neuronas	LeakyReLU	
Capa 7	Salida	3×3		SoftMax	3×3

Una vez realizado el entrenamiento y la validación de esta arquitectura propuesta en conjunto con los hiperparámetros mostrados en la tabla 3.72, se obtienen los siguientes resultados:

Tabla 3.72: Hiperparámetros de la Red Neuronal Convolutiva Propuesta 5

Hiperparámetros	
Optimizador	RMSprop
Función de pérdidas	CATEGORICAL CROSSENTROPY
Tasa de aprendizaje	0.001

- Matriz de confusión

Tabla 3.73: Matriz de Confusión Arquitectura Propuesta 5

	Covid-19	Neumonía	Normal
Covid-19	173	0	13
Neumonía	19	40	16
Normal	65	1	432

- Porcentaje de Error y Exactitud

Tabla 3.74: Porcentaje de Error y exactitud Arquitectura Propuesta 5

Arquitectura	Error mínimo entrenamiento	Accuracy Máximo Entrenamiento	Error mínimo Prueba	Accuracy Máximo Prueba
Libre 5	0.1109 %	97.2783 %	0.5440 %	90.6455 %

- Porcentaje de Precisión, Sensibilidad y F1 Score

Tabla 3.75: Métricas Arquitectura Propuesta 5

	Covid-19	Neumonía	Normal
Precisión	67.3151 %	97.5609 %	93.7093 %
Sensibilidad	93.0107 %	53.3333 %	86.7469 %
F1 Score	78.1038 %	68.9655 %	90.0938 %

3.3. Arquitecturas Conocidas

Las arquitecturas conocidas hacen referencia a arquitecturas conocidas y utilizadas mayormente en trabajos similares con muy buenos resultados; estas fueron implementadas con el fin de comprobar la efectividad de diferentes arquitecturas conocidas comparadas con las arquitecturas libres. Estas arquitecturas tienen en común la gran extensión y complejidad de sus capas.

Como primera instancia, se realizan pruebas con diferentes valores de separación de la base de datos; es decir, se proponen 3 diferentes métodos de separación, el primero con 70 % de los datos para entrenamiento y 30 % para validación, la segunda propuesta es de 80 % para entrenamiento y 20 % para validación, y la última propuesta es de 95 % para entrenamiento y 5 % para validación.

Partiendo de esto, a continuación se muestran los resultados obtenidos en esta primera prueba.

Estas arquitecturas se implementaron como comparativos debido a sus aportes en diferentes trabajos como [29]. que muestra el avance en el ámbito médico empleando las redes neuronales convolucionales Lenet 1.

3.3.1. Selección 70/30

LeNet 1

Tabla 3.76: Estructura de la Red Neuronal Convolutiva LeNet 1

	Nombre	Estructura	Valor	Función de Activación	Tamaño del Kernel
Capa 1	Entrada	256×256	32 filtros	ReLu Glorot Uniforme	3×3
	Convolución 1 Kernel de Inicialización	256×256			
Capa 2	Max Pooling 1	2×2			
Capa 3	Convolución 2	128×128	8 filtros	ReLu Glorot Uniforme	5×5
	Kernel de Inicialización				
Capa 4	Average Pooling 1	2×2			
Capa 5	Full Conected				
Capa 6	Salida	3×3		SoftMax	3×3

Una vez realizado el entrenamiento y la validación de esta arquitectura propuesta en conjunto con los hiperparámetros mostrados en la tabla 3.77, se obtienen los siguientes resultados:

Tabla 3.77: Hiperparámetros de la Red Neuronal Convolutiva LeNet 1

Hiperparámetros	
Optimizador	ADAM
Función de pérdidas	CATEGORICAL CROSSENTROPY
Tasa de aprendizaje	0.001

Tabla 3.78: Matriz de Confusión Arquitectura LeNet 1

	Covid-19	Neumonía	Normal
Covid-19	929	7	170
Neumonía	20	365	29
Normal	117	21	2.901

- Porcentaje de Error y Exactitud

Tabla 3.79: Porcentaje de Error y exactitud Arquitectura LeNet 1

Arquitectura	Error mínimo entrenamiento	Accuracy Máximo Entrenamiento	Error mínimo Prueba	Accuracy Máximo Prueba
LeNet 1	0.1112 %	96.0150 %	0.2136 %	92.4983 %

- Porcentaje de Precisión, Sensibilidad y F1 Score

Tabla 3.80: Métricas Arquitectura LeNet 1

	Covid-19	Neumonía	Normal
Precisión	87.1482 %	92.8753 %	93.5806 %
Sensibilidad	83.9963 %	88.1642 %	95.4590 %
F1 Score	85.5432 %	90.4584 %	94.5105 %

LeNet 1 con Variación**Tabla 3.81:** Estructura de la Red Neuronal Convolutiva LeNet 1 con Variación

	Nombre	Estructura	Valor	Función de Activación	Tamaño del Kernel
Capa 1	Entrada				
	Convolución 1 Kernel de Inicialización	256×256 256×256	32 filtros	LeakyReLu Glorot Normal	3×3
Capa 2	Max Pooling 1	2×2			
Capa 3	Convolución 2	128×128	8 filtros	LeakyReLu	5×5
	Kernel de Inicialización			Glorot Normal	
Capa 4	Average Pooling 1	2×2			
Capa 5	Full Conected				
Capa 6	Salida	3×3		SoftMax	3×3

Una vez realizado el entrenamiento y la validación de esta arquitectura propuesta en conjunto con los hiperparámetros mostrados en la tabla 3.82, se obtienen los siguientes resultados:

Tabla 3.82: Hiperparámetros de la Red Neuronal Convolutiva LeNet 1 con Variación

Hiperparámetros	
Optimizador	NADAM
Función de pérdidas	CATEGORICAL CROSSENTROPY
Tasa de aprendizaje	0.001

- Matriz de confusión

Tabla 3.83: Matriz de Confusión Arquitectura LeNet 1.2

	Covid-19	Neumonía	Normal
Covid-19	868	7	231
Neumonía	24	354	36
Normal	99	21	2.919

- Porcentaje de Error y Exactitud

Tabla 3.84: Porcentaje de Error y exactitud Arquitectura LeNet 1 con Variación

Arquitectura	Error mínimo entrenamiento	Accuracy Máximo Entrenamiento	Error mínimo Prueba	Accuracy Máximo Prueba
LeNet 1.2	0.2240 %	90.8313 %	0.2532 %	91.7011 %

- Porcentaje de Precisión, Sensibilidad y F1 Score

Tabla 3.85: Métricas Arquitectura LeNet 1 con Variación

	Covid-19	Neumonía	Normal
Precisión	87.5829 %	92.6701 %	91.6195 %
Sensibilidad	78.4810 %	85.5072 %	96.0513 %
F1 Score	82.7849 %	88.9447 %	93.7813 %

LeNet5

Tabla 3.86: Estructura de la Red Neuronal Convolutiva LeNet 5

	Nombre	Estructura	Valor	Función de Activación	Tamaño del Kernel
Capa 1	Entrada				
	Convolución 1 Kernel de Inicialización	256×256 256×256	6 filtros	TanH Glorot Uniforme	5×5
Capa 2	Max Pooling 1	2×2			
Capa 3	Convolución 2				
	Kernel de Inicialización	128×128	16 filtros	TanH Glorot Uniforme	5×5
Capa 4	Max Pooling 2	2×2			
Capa 5	Full Conected				
Capa 6	Densa		182 Neuronas	TanH	
Capa 7	Densa		182 Neuronas	TanH	
Capa 8	Salida	3×3		SoftMax	3×3

Una vez realizado el entrenamiento y la validación de esta arquitectura propuesta en conjunto con los hiperparámetros mostrados en la tabla 3.87, se obtienen los siguientes resultados:

Tabla 3.87: Hiperparámetros de la Red Neuronal Convolutiva LeNet 1 con Variación

Hiperparámetros	
Optimizador	ADAM
Función de pérdidas	CATEGORICAL CROSSENTROPY
Tasa de aprendizaje	0.001

- Matriz de confusión

Tabla 3.88: Matriz de Confusión Arquitectura LeNet 5

	Covid-19	Neumonía	Normal
Covid-19	0	0	1.106
Neumonía	0	0	414
Normal	0	0	3.039

- Porcentaje de Error y Exactitud

Tabla 3.89: Porcentaje de Error y exactitud Arquitectura LeNet 5

Arquitectura	Error mínimo entrenamiento	Accuracy Máximo Entrenamiento	Error mínimo Prueba	Accuracy Máximo Prueba
LeNet 5	0.8304 %	66.6593 %	0.8326 %	67.2368 %

- Porcentaje de Precisión, Sensibilidad y F1 Score

Tabla 3.90: Métricas Arquitectura LeNet 5

	Covid-19	Neumonía	Normal
Precisión	–	–	66.6593 %
Sensibilidad	0 %	0 %	100 %
F1 Score	–	–	79.9947 %

LeNet5 con Variación

Tabla 3.91: Estructura de la Red Neuronal Convolutiva LeNet 5 con Variación

	Nombre	Estructura	Valor	Función de Activación	Tamaño del Kernel
Capa 1	Entrada	256×256	6 filtros	LeakyReLU Glorot Normal	5×5
	Convolución 1 Kernel de Inicialización	256×256			
Capa 2	Max Pooling 1	2×2			
Capa 3	Convolución 2	128×128	16 filtros	LeakyReLU Glorot Normal	5×5
	Kernel de Inicialización				
Capa 4	Max Pooling 2	2×2			
Capa 5	Full Conected				
Capa 6	Densa		182 Neuronas	LeakyReLU	
Capa 7	Densa		182 Neuronas	LeakyReLU	
Capa 8	Salida	3×3		SoftMax	3×3

Una vez realizado el entrenamiento y la validación de esta arquitectura propuesta en conjunto con los hiperparámetros mostrados en la tabla 3.92, se obtienen los siguientes resultados:

Tabla 3.92: Hiperparámetros de la Red Neuronal Convolutiva LeNet 1 con Variación

Hiperparámetros	
Optimizador	NADAM
Función de pérdidas	CATEGORICAL CROSSENTROPY
Tasa de aprendizaje	0.001

- Matriz de confusión

Tabla 3.93: Matriz de Confusión Arquitectura LeNet 5.2

	Covid-19	Neumonía	Normal
Covid-19	1.019	5	82
Neumonía	18	370	26
Normal	118	22	2.899

- Porcentaje de Error y Exactitud

Tabla 3.94: Porcentaje de Error y exactitud Arquitectura LeNet 5.2

Arquitectura	Error mínimo entrenamiento	Accuracy Máximo Entrenamiento	Error mínimo Prueba	Accuracy Máximo Prueba
LeNet 5	0.0790 %	97.2556 %	0.1773 %	94.0557 %

- Porcentaje de Precisión, Sensibilidad y F1 Score

Tabla 3.95: Métricas Arquitectura LeNet 5 con Variación

	Covid-19	Neumonía	Normal
Precisión	88.2251 %	93.1989 %	96.4083 %
Sensibilidad	92.1338 %	89.3719 %	95.3922 %
F1 Score	90.1371 %	91.2453 %	95.8981 %

3.3.2. Selección 80/20

LeNet 1

Tabla 3.96: Estructura de la Red Neuronal Convolutacional LeNet 1

	Nombre	Estructura	Valor	Función de Activación	Tamaño del Kernel
Capa 1	Entrada	256×256	32 filtros	ReLu Glorot Uniforme	3×3
	Convolución 1 Kernel de Inicialización	256×256			
Capa 2	Max Pooling 1	2×2			
Capa 3	Convolución 2	128×128	8 filtros	ReLu Glorot Uniforme	5×5
	Kernel de Inicialización				
Capa 4	Average Pooling 1	2×2			
Capa 5	Full Conected				
Capa 6	Salida	3×3		SoftMax	3×3

Una vez realizado el entrenamiento y la validación de esta arquitectura propuesta en conjunto con los hiperparámetros mostrados en la tabla 3.97, se obtienen los siguientes resultados:

Tabla 3.97: Hiperparámetros de la Red Neuronal Convolutacional LeNet 1

Hiperparámetros	
Optimizador	ADAM
Función de pérdidas	CATEGORICAL CROSSENTROPY
Tasa de aprendizaje	0.001

- Matriz de confusión

Tabla 3.98: Matriz de Confusión Arquitectura LeNet 1

	Covid-19	Neumonía	Normal
Covid-19	684	4	65
Neumonía	5	252	14
Normal	73	16	1.926

- Porcentaje de Error y Exactitud

Tabla 3.99: Porcentaje de Error y exactitud Arquitectura LeNet 1

Arquitectura	Error mínimo entrenamiento	Accuracy Máximo Entrenamiento	Error mínimo Prueba	Accuracy Máximo Prueba
LeNet 1	0.0841 %	97.0312 %	0.1850 %	94.1757 %

- Porcentaje de Precisión, Sensibilidad y F1 Score

Tabla 3.100: Métricas Arquitectura LeNet 1

	Covid-19	Neumonía	Normal
Precisión	89.7637 %	92.6470 %	96.0598 %
Sensibilidad	90.8366 %	92.9889 %	95.5831 %
F1 Score	90.2970 %	92.8176 %	95.8208 %

LeNet 1 con Variación

Tabla 3.101: Estructura de la Red Neuronal Convolutiva LeNet 1 con Variación

	Nombre	Estructura	Valor	Función de Activación	Tamaño del Kernel
Capa 1	Entrada				
	Convolución 1 Kernel de Inicialización	256×256 256×256	32 filtros	LeakyReLU Glorot Normal	3×3
Capa 2	Max Pooling 1	2×2			
Capa 3	Convolución 2	128×128	8 filtros	LeakyReLU	5×5
	Kernel de Inicialización			Glorot Normal	
Capa 4	Average Pooling 1	2×2			
Capa 5	Full Conected				
Capa 6	Salida	3×3		SoftMax	3×3

Una vez realizado el entrenamiento y la validación de esta arquitectura propuesta en conjunto con los hiperparámetros mostrados en la tabla 3.102, se obtienen los siguientes resultados:

Tabla 3.102: Hiperparámetros de la Red Neuronal Convolutiva LeNet 1 con Variación

Hiperparámetros	
Optimizador	NADAM
Función de pérdidas	CATEGORICAL CROSSENTROPY
Tasa de aprendizaje	0.001

- Matriz de confusión

Tabla 3.103: Matriz de Confusión Arquitectura LeNet 1.2

	Covid-19	Neumonía	Normal
Covid-19	658	2	93
Neumonía	16	237	18
Normal	151	18	1.846

- Porcentaje de Error y Exactitud

Tabla 3.104: Porcentaje de Error y exactitud Arquitectura LeNet 1 con Variación

Arquitectura	Error mínimo entrenamiento	Accuracy Máximo Entrenamiento	Error mínimo Prueba	Accuracy Máximo Prueba
LeNet 1.2	0.2145 %	92.1052 %	0.2620 %	90.3586 %

- Porcentaje de Precisión, Sensibilidad y F1 Score

Tabla 3.105: Métricas Arquitectura LeNet 1 con Variación

	Covid-19	Neumonía	Normal
Precisión	79.7575 %	92.2178 %	94.3280 %
Sensibilidad	87.3837 %	87.4538 %	91.6129 %
F1 Score	83.3967 %	89.7727 %	92.5065 %

LeNet5

Tabla 3.106: Estructura de la Red Neuronal Convolutiva LeNet 5

	Nombre	Estructura	Valor	Función de Activación	Tamaño del Kernel
Capa 1	Entrada	256×256	6 filtros	TanH	5×5
	Convolución 1 Kernel de Inicialización	256×256		Glorot Uniforme	
Capa 2	Max Pooling 1	2×2			
Capa 3	Convolución 2	128×128	16 filtros	TanH	5×5
	Kernel de Inicialización			Glorot Uniforme	
Capa 4	Max Pooling 2	2×2			
Capa 5	Full Conected				
Capa 6	Densa		182 Neuronas	TanH	
Capa 7	Densa		182 Neuronas	TanH	
Capa 8	Salida	3×3		SoftMax	3×3

Una vez realizado el entrenamiento y la validación de esta arquitectura propuesta en conjunto con los hiperparámetros mostrados en la tabla 3.107, se obtienen los siguientes resultados:

Tabla 3.107: Hiperparámetros de la Red Neuronal Convolutiva LeNet 1 con Variación

Hiperparámetros	
Optimizador	ADAM
Función de pérdidas	CATEGORICAL CROSSENTROPY
Tasa de aprendizaje	0.001

- Matriz de confusión

Tabla 3.108: Matriz de Confusión Arquitectura LeNet 5

	Covid-19	Neumonía	Normal
Covid-19	0	0	753
Neumonía	0	0	271
Normal	0	0	2.015

- Porcentaje de Error y Exactitud

Tabla 3.109: Porcentaje de Error y exactitud Arquitectura LeNet 5

Arquitectura	Error mínimo entrenamiento	Accuracy Máximo Entrenamiento	Error mínimo Prueba	Accuracy Máximo Prueba
LeNet 5	0.8229 %	67.2532 %	0.8337 %	66.3047 %

- Porcentaje de Precisión, Sensibilidad y F1 Score

Tabla 3.110: Métricas Arquitectura LeNet 5

	Covid-19	Neumonía	Normal
Precisión	–	–	66.3047 %
Sensibilidad	0 %	0 %	100 %
F1 Score	–	–	79.7388 %

LeNet5 con Variación

Tabla 3.111: Estructura de la Red Neuronal Convolutacional LeNet 5 con Variación

	Nombre	Estructura	Valor	Función de Activación	Tamaño del Kernel
Capa 1	Entrada	256×256	6 filtros	LeakyReLU Glorot Normal	5×5
	Convolución 1 Kernel de Inicialización	256×256			
Capa 2	Max Pooling 1	2×2			
Capa 3	Convolución 2	128×128	16 filtros	LeakyReLU Glorot Normal	5×5
	Kernel de Inicialización				
Capa 4	Max Pooling 2	2×2			
Capa 5	Full Conected				
Capa 6	Densa		182 Neuronas	LeakyReLU	
Capa 7	Densa		182 Neuronas	LeakyReLU	
Capa 8	Salida	3×3		SoftMax	3×3

Una vez realizado el entrenamiento y la validación de esta arquitectura propuesta en conjunto con los hiperparámetros mostrados en la tabla 3.112, se obtienen los siguientes resultados:

Tabla 3.112: Hiperparámetros de la Red Neuronal Convolutacional LeNet 1 con Variación

Hiperparámetros	
Optimizador	NADAM
Función de pérdidas	CATEGORICAL CROSSENTROPY
Tasa de aprendizaje	0.001

- Matriz de confusión

Tabla 3.113: Matriz de Confusión Arquitectura LeNet 5.2

	Covid-19	Neumonía	Normal
Covid-19	658	1	67
Neumonía	17	228	26
Normal	105	8	1.902

- Porcentaje de Error y Exactitud

Tabla 3.114: Porcentaje de Error y exactitud Arquitectura LeNet 5.2

Arquitectura	Error mínimo entrenamiento	Accuracy Máximo Entrenamiento	Error mínimo Prueba	Accuracy Máximo Prueba
LeNet 5	0.1535 %	94.5559 %	0.1995 %	92.6291 %

- Porcentaje de Precisión, Sensibilidad y F1 Score

Tabla 3.115: Métricas Arquitectura LeNet 5 con Variación

	Covid-19	Neumonía	Normal
Precisión	84.8822 %	96.2025 %	95.3383 %
Sensibilidad	90.9694 %	84.1328 %	94.3920 %
F1 Score	87.8205 %	89.7637 %	94.8628 %

3.3.3. Selección 95/5

LeNet 1

Tabla 3.116: Estructura de la Red Neuronal Convolutiva LeNet 1

	Nombre	Estructura	Valor	Función de Activación	Tamaño del Kernel
Capa 1	Entrada	256×256	32 filtros	ReLu	3×3
	Convolución 1 Kernel de Inicialización	256×256			
Capa 2	Max Pooling 1	2×2		Glorot Uniforme	
Capa 3	Convolución 2	128×128	8 filtros	ReLu	5×5
	Kernel de Inicialización				
Capa 4	Average Pooling 1	2×2			
Capa 5	Full Conected				
Capa 6	Salida	3×3		SoftMax	3×3

Una vez realizado el entrenamiento y la validación de esta arquitectura propuesta en conjunto con los hiperparámetros mostrados en la tabla 3.117, se obtienen los siguientes resultados:

Tabla 3.117: Hiperparámetros de la Red Neuronal Convolutiva LeNet 1

Hiperparámetros	
Optimizador	ADAM
Función de pérdidas	CATEGORICAL CROSSENTROPY
Tasa de aprendizaje	0.001

- Matriz de confusión

Tabla 3.118: Matriz de Confusión Arquitectura LeNet 1

	Covid-19	Neumonía	Normal
Covid-19	167	2	17
Neumonía	1	67	7
Normal	12	6	480

- Porcentaje de Error y Exactitud

Tabla 3.119: Porcentaje de Error y exactitud Arquitectura LeNet 1

Arquitectura	Error mínimo entrenamiento	Accuracy Máximo Entrenamiento	Error mínimo Prueba	Accuracy Máximo Prueba
LeNet 1	0.0777 %	97.2714 %	0.2048 %	94.0711 %

- Porcentaje de Precisión, Sensibilidad y F1 Score

Tabla 3.120: Métricas Arquitectura LeNet 1

	Covid-19	Neumonía	Normal
Precisión	92.7777 %	89.3333 %	95.2380 %
Sensibilidad	89.7849 %	89.3333 %	96.3835 %
F1 Score	91.2568 %	89.3333 %	95.8083 %

LeNet 1 con Variación

Tabla 3.121: Estructura de la Red Neuronal Convolutiva LeNet 1 con Variación

	Nombre	Estructura	Valor	Función de Activación	Tamaño del Kernel
Capa 1	Entrada	256×256	32 filtros	LeakyReLU Glorot Normal	3×3
	Convolución 1 Kernel de Inicialización	256×256			
Capa 2	Max Pooling 1	2×2			
Capa 3	Convolución 2	128×128	8 filtros	LeakyReLU Glorot Normal	5×5
	Kernel de Inicialización				
Capa 4	Average Pooling 1	2×2			
Capa 5	Full Conected				
Capa 6	Salida	3×3		SoftMax	3×3

Una vez realizado el entrenamiento y la validación de esta arquitectura propuesta en conjunto con los hiperparámetros mostrados en la tabla 3.122, se obtienen los siguientes resultados:

Tabla 3.122: Hiperparámetros de la Red Neuronal Convolutiva LeNet 1 con Variación

Hiperparámetros	
Optimizador	NADAM
Función de pérdidas	CATEGORICAL CROSSENTROPY
Tasa de aprendizaje	0.001

- Matriz de confusión

Tabla 3.123: Matriz de Confusión Arquitectura LeNet 1.2

	Covid-19	Neumonía	Normal
Covid-19	146	1	39
Neumonía	3	64	8
Normal	7	2	489

- Porcentaje de Error y Exactitud

Tabla 3.124: Porcentaje de Error y exactitud Arquitectura LeNet 1 con Variación

Arquitectura	Error mínimo entrenamiento	Accuracy Máximo Entrenamiento	Error mínimo Prueba	Accuracy Máximo Prueba
LeNet 1.2	0.1813 %	93.5590 %	0.2471 %	92.0948 %

- Porcentaje de Precisión, Sensibilidad y F1 Score

Tabla 3.125: Métricas Arquitectura LeNet 1 con Variación

	Covid-19	Neumonía	Normal
Precisión	93.5897 %	95.5223 %	91.2313 %
Sensibilidad	78.4962 %	85.3333 %	98.1927 %
F1 Score	85.3801 %	90.1408 %	94.5841 %

LeNet5

Tabla 3.126: Estructura de la Red Neuronal Convolutacional LeNet 5

	Nombre	Estructura	Valor	Función de Activación	Tamaño del Kernel
Capa 1	Entrada	256×256	6 filtros	TanH	5×5
	Convolución 1 Kernel de Inicialización	256×256			
Capa 2	Max Pooling 1	2×2			
Capa 3	Convolución 2	128×128	16 filtros	TanH	5×5
	Kernel de Inicialización				
Capa 4	Max Pooling 2	2×2			
Capa 5	Full Conected				
Capa 6	Densa		182 Neuronas	TanH	
Capa 7	Densa		182 Neuronas	TanH	
Capa 8	Salida	3×3		SoftMax	3×3

Una vez realizado el entrenamiento y la validación de esta arquitectura propuesta en conjunto con los hiperparámetros mostrados en la tabla 3.127, se obtienen los siguientes resultados:

Tabla 3.127: Hiperparámetros de la Red Neuronal Convolutacional LeNet 1 con Variación

Hiperparámetros	
Optimizador	ADAM
Función de pérdidas	CATEGORICAL CROSSENTROPY
Tasa de aprendizaje	0.001

- Matriz de confusión

Tabla 3.128: Matriz de Confusión Arquitectura LeNet 5

	Covid-19	Neumonía	Normal
Covid-19	0	0	186
Neumonía	0	0	75
Normal	0	0	498

- Porcentaje de Error y Exactitud

Tabla 3.129: Porcentaje de Error y exactitud Arquitectura LeNet 5

Arquitectura	Error mínimo entrenamiento	Accuracy Máximo Entrenamiento	Error mínimo Prueba	Accuracy Máximo Prueba
LeNet 5	0.8297 %	67.1398 %	0.85 %	65.6126 %

- Porcentaje de Precisión, Sensibilidad y F1 Score

Tabla 3.130: Métricas Arquitectura LeNet 5

	Covid-19	Neumonía	Normal
Precisión	–	–	65.6126 %
Sensibilidad	0 %	0 %	100 %
F1 Score	–	–	79.2362 %

LeNet5 con Variación**Tabla 3.131:** Estructura de la Red Neuronal Convolutiva LeNet 5 con Variación

	Nombre	Estructura	Valor	Función de Activación	Tamaño del Kernel
Capa 1	Entrada	256 × 256	6 filtros	LeakyReLU Glorot Normal	5 × 5
	Convolución 1 Kernel de Inicialización	256 × 256			
Capa 2	Max Pooling 1	2 × 2			
Capa 3	Convolución 2	128 × 128	16 filtros	LeakyReLU Glorot Normal	5 × 5
	Kernel de Inicialización				
Capa 4	Max Pooling 2	2 × 2			
Capa 5	Full Conected				
Capa 6	Densa		182 Neuronas	LeakyReLU	
Capa 7	Densa		182 Neuronas	LeakyReLU	
Capa 8	Salida	3 × 3		SoftMax	3 × 3

Una vez realizado el entrenamiento y la validación de esta arquitectura propuesta en conjunto con los hiperparámetros mostrados en la tabla 3.132, se obtienen los siguientes resultados:

Tabla 3.132: Hiperparámetros de la Red Neuronal Convolutiva LeNet 1 con Variación

Hiperparámetros	
Optimizador	NADAM
Función de pérdidas	CATEGORICAL CROSSENTROPY
Tasa de aprendizaje	0.001

- Matriz de confusión

Tabla 3.133: Matriz de Confusión Arquitectura LeNet 5.2

	Covid-19	Neumonía	Normal
Covid-19	154	4	28
Neumonía	1	67	7
Normal	6	1	491

- Porcentaje de Error y Exactitud

Tabla 3.134: Porcentaje de Error y exactitud Arquitectura LeNet 5.2

Arquitectura	Error mínimo entrenamiento	Accuracy Máximo Entrenamiento	Error mínimo Prueba	Accuracy Máximo Prueba
LeNet 5	0.0965 %	96.6481 %	0.1851 %	94.3346 %

- Porcentaje de Precisión, Sensibilidad y F1 Score

Tabla 3.135: Métricas Arquitectura LeNet 5 con Variación

	Covid-19	Neumonía	Normal
Precisión	95.6521 %	93.0555 %	93.3460 %
Sensibilidad	82.7956 %	89.3333 %	98.5943 %
F1 Score	88.7608 %	91.1564 %	95.8984 %

Una vez realizada la primera etapa de prueba, se realizó una selección de las mejores arquitecturas, en la cual se escogen las mejores arquitecturas tanto de la libre como de las conocidas; de igual manera para la separación de 70/30 como de 80/20; basándose en el resultado obtenido de la métrica F1 Score, ya que esta combina la sensibilidad y precisión, por ende podemos saber que tan robusto puede llegar a ser el modelo a implementar; esto con el fin de realizar diferentes modificaciones, tales como: función de activación (Filtro 1), Optimizador (Filtro 2) y Kernel de inicialización (Filtro 3).

Para el Filtro 1 se tomaron de la separación 70/30 las arquitecturas Libre 5 y LeNet 5 con variación; de la separación 80/20 se tomaron las arquitecturas Libre 2 y LeNet 1; se evidencian en los resultados obtenidos de las tablas 3.25, 3.95, 3.35 y 3.100 respectivamente.

3.4. Filtro 1: Modificación de la función de Activación

Una vez realizadas todas las modificaciones propuestas para el filtro 1; las cuales fueron modificación de las funciones de activación entre Relu, LeakyRelu, Sigmoid y Tangente Hiperbólica, se escogen las arquitecturas con mejores resultados, tanto de la separación 70/30 como la del 80/20.

En cuanto a la separación 70/30 se obtuvo mejoras en la arquitectura LeNet 5.2 en un 1.994 % y para la separación 80/20 la arquitectura Libre 2 en un 2.8964 %; en las arquitecturas Libre 5 de la separación 70/30 y LeNet 1 de la separación 80/20 no se evidenciaron mejoras por lo que se dejan tal cual como se tenían en los resultados de la prueba 1.

Para el siguiente filtro, se toman en cuenta el análisis hecho en el filtro 1 y se realizan las modificaciones tanto de la función de activación como del tamaño del lote (batch) para las arquitecturas que tuvieron mejoras y se procede a realizar pruebas donde se modifique el hiperparámetro denominado como optimizador, esto con el fin de ver si se puede lograr mejores resultados.

3.5. Filtro 2: Modificación del optimizador

Una vez realizado realizadas todas las modificaciones propuestas para el filtro 2; las cuales fueron modificación del optimizador entre Adam Nadam, Adagrad, RMSprop, AdaDelta; se escogen las arquitecturas con mejores resultados, tanto de la separación 70/30 como la del 80/20.

En cuanto a la separación 70/30 se obtuvo mejoras en la arquitectura Libre 5 con el optimizador Adagrad en un 7,5052 % y para la separación 80/20 la arquitectura Libre 2 con el optimizador RMSProp en un 0,87242 %; en las arquitecturas LeNet 5.2 de la separación 70/30 y LeNet 1 de la separación 80/20 no se evidenciaron mejoras por lo que se dejan tal cual como se tenían en los resultados del filtro 1.

Para el siguiente filtro, se toman en cuenta el análisis hecho en el filtro 2 y se realizan las modificaciones en los optimizadores para las arquitecturas que tuvieron mejoras y se procede a realizar pruebas donde se modifique el parámetro del kernel de iniciación, esto con el fin de ver si se puede lograr mejores resultados.

3.6. Filtro 3: Modificación Kernel de Inicialización

Una vez realizado realizadas todas las modificaciones propuestas para el filtro 3; las cuales fueron modificación del Kernel de Inicialización entre Glorot Uniforme, Glorot Normal, He Uniforme, He Normal, y Ortogonal; se escogen las arquitecturas con mejores resultados, tanto de la separación 70/30 como la del 80/20.

En cuanto a la separación 70/30 se obtuvo mejoras en la arquitectura LeNet 5 con variación, función de activación ReLu, optimizador Nadam y kernel de inicialización Glorot Normal en un 3,0119 % y para la separación 80/20 la arquitectura LeNet 1 con función de activación ReLu, optimizador Adam y kernel de inicialización He Normal en un 3,5446 %; en las arquitecturas.

- Porcentaje de F1 Score Para 70/30

Tabla 3.136: Porcentaje de F1 Score Arquitectura LeNet 5 con Variación

Arquitectura	F1 Score Covid	F1 Score Neumonía	F1 Score Normal
LeNet 5.2	90.6923 %	91.1951 %	96.3871 %

- Porcentaje de F1 Score Para 80/20

Tabla 3.137: Porcentaje de F1 Score Arquitectura LeNet 1

Arquitectura	F1 Score Covid	F1 Score Neumonía	F1 Score Normal
LeNet 5.2	89.6167 %	93.9671 %	95.9473

Conclusiones

En este trabajo, se presenta un análisis variado que involucran diferentes tipos de hiperparámetros en diferentes redes convolucionales, con el fin de determinar el impacto con cada combinación y en cada etapa.

La metodología plantea la clasificación de imágenes de rayos X pulmonares para determinar 3 tipos de enfermedades determinadas desde el inicio del proyecto. Cabe resaltar que las 3 clases no están balanceadas en número de imágenes; por ende, se tiene en cuenta ese factor a la hora de analizar todas métricas de valoración al final.

Para el desarrollo de la metodología, se proponen 9 diferentes arquitecturas de redes neuronales convolucionales, esto con la idea de realizar una comparativa extensa y poder dar a conocer los diferentes resultados en cada etapa de modificación; a su vez se toma como primordial la métrica F1 Score la cual involucra la precisión y la exhaustividad.

Los resultados numéricos obtenidos por las métricas de F1 Score en la separación de 70/30 y 80/20 demuestran las mejoras obtenidas a lo largo de las modificaciones hechas en todos los filtros planteados, y permiten un análisis de la sensibilidad de los datos en diferentes circunstancias. Adicionalmente a ello, el sin número de combinaciones en las capas de las arquitecturas que se pueden proponer son infinitas.

Para el caso de la partición 70/30, la arquitectura con mejores resultados luego de aplicados todos los filtros propuestos es la arquitectura LeNet 5.2 la cual es la LeNet 5 con Variación y entre sus hiperparámetros tiene función de activación ReLu, optimizador Nadam, Kernel de inicialización Glorot Normal y batch size de 250; y para el caso de la partición 80/20 la arquitectura con mejores resultados luego de aplicados todos los filtros propuestos es la arquitectura LeNet 1, la cual tiene como hiperparámetros función de activación ReLu, optimizador Adam, Kernel de inicialización He Normal y batch size de 100.

Para poder llegar a estos datos mencionados anteriormente se realizó como primera medida una prueba con la técnica de clasificación de imágenes propuesta por Haralick; esto con el fin de filtrar atributos de la base de datos, sabiendo que por este método se podría analizar los descriptores de texturas de una imagen mediante una matriz de co-ocurrencia de niveles de grises o con sus siglas en ingles GLCM (Grey Level Co-ocurrence Matrix). Se observó que el tiempo de procesamiento y el resultado obtenido en cuanto a la clasificación de

imágenes no era satisfactorio, por ende, Finalizando dicho análisis se obtuvo un 72.44 % de exactitud máxima probando con 1.000 épocas de iteraciones con un tiempo aproximado de 32 minutos en ejecución, por lo cual se decidió realizar un filtrado de atributos mediante redes neuronales convolucionales.

Una vez escogidas las redes neuronales convolucionales como método de filtrado atributos dentro de la clasificación de imágenes, se adaptaron técnicas de procesamiento digital de imágenes con el fin de estandarizar y/o acondicionar las imágenes de la base de datos, para esto se realizó una normalización la cual consistió en el escalamiento de las imágenes de entrada, teniendo como imagen de entrada un tamaño de 256x256x3. Cabe resaltar que la forma de filtración de datos se realiza por medio de un vector categórico, el cual se utiliza para almacenar datos de valores provenientes de un conjunto finito de categorías discretas.

Estandarizadas las imágenes, se procede a realizar el entrenamiento y posterior validación de las 9 diferentes arquitecturas con hiperparámetros iniciales, y sus diferentes filtros y/o modificaciones en cada etapa; la validación se realizó mediante el descarte por mejoras en la métrica F1 Score y el tiempo de ejecución.

Al terminar las diferentes etapas, filtrados y ajustes en los hiperparámetros, se pudo concluir que al plantear 9 diferentes arquitecturas de redes neuronales convolucionales, la mejor para este caso en particular es Arquitectura LeNet 5.2 de la partición 70/30, función de activación ReLu, optimizador Nadam, kernel de inicialización Glorot Normal y Batch Size de 250 como se evidencia en los resultados obtenidos de la tabla 3.136.

Trabajos futuros

Con respecto a la investigación desarrollada hasta el momento, se proponen los siguientes temas como posibles trabajos futuros:

- Realizar un balance en la base de datos
- Implementar nuevas combinaciones de los hiperparámetros en las arquitecturas con mejores resultados

Referencias

- [1] Juan Bagnato. ¿cómo funcionan las convolutional neural networks? visión por ordenador, Nov 2018. URL <https://www.aprendemachinelearning.com/como-funcionan-las-convolutional-neural-networks-vision-por-ordenador/>.
- [2] Diego Calvo. Función de activación – redes neuronales, Dec 2018. URL <https://www.diegocalvo.es/funcion-de-activacion-redes-neuronales/#:~:text=Sigmoid20E2809320Sigmoide,de20manera20asintC3B3tica20a200.&text=CaracterC3ADsticas20de20la20funciC3B3n20signoide,Lenta20convergencia.>
- [3] Instituto Nacional del Cáncer de los Institutos Nacionales de la Salud de EE. UU. Diccionario de cáncer del nci, 2010. URL <https://www.cancer.gov/espanol/publicaciones/diccionarios/diccionario-cancer/def/neurona>.
- [4] Instituto Nacional del Cáncer de los Institutos Nacionales de la Salud de EE. UU. Diccionario de cáncer del nci, 2010. URL <https://www.cancer.gov/espanol/publicaciones/diccionarios/diccionario-cancer/def/neurona>.
- [5] ESTHER DORADO. *DISEÑO, IMPLEMENTACIÓN Y EVALUACIÓN DE UNA RED NEURONAL CONVOLUCIONAL PARA LA DETECCIÓN DE PUNTOS PRINCIPALES EN NARANJAS*. PhD thesis, Universidad Politécnica de Valencia, 2020. URL riunet.upv.es/bitstream/handle/10251/162073/72897452X_TFM_16002735057456800632002134776110.pdf?sequence=1.
- [6] Jaime Durán. Técnicas de regularización básicas para redes neuronales, Oct 2019. URL <https://medium.com/metadatos/t%C3%A9cnicas-de-regularizaci%C3%B3n-b%C3%A1sicas-para-redes-neuronales-b48f396924d4>.
- [7] Chollet François. *Deep learning with python*. Manning Publications Co., 2018.
- [8] José L. Fresquet. Historia de la primera radiografía, May 2022. URL <https://www.foronuclear.org/actualidad/a-fondo/historia-de-la-primera-radiografia/>.
- [9] Salvador Gil, Silvia Calderón, Pablo Núñez, Di Laccio, and Leila Iannelli. Aulas-laboratorios de bajo costo, usando tic. *Revista Eureka sobre enseñanza y divulgación de las ciencias.*, 12:212–226, 01 2015. doi: 10.25267/Rev_Eureka_ensen_divulg_cienc.2015.v12.i1.15.
- [10] Dmitriy Gizlyk. Redes neuronales: Así de sencillo (parte 12): Dropout, May 2021. URL <https://www.mql5.com/es/articles/9112>.

- [11] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *JMLR Workshop and Conference Proceedings*, pages 249–256, 2010.
- [12] Ceridwen Johnson. Las nuevas directrices mundiales de la oms sobre la calidad del aire tienen como objetivo evitar millones de muertes debidas a la contaminación del aire, 2021. URL <https://www.who.int/es/news/item/22-09-2021-new-who-global-air-quality-guidelines-aim-to-save-millions-of-lives-from-air-pollution>.
- [13] Ceridwen Johnson. Coronavirus, 2021. URL https://www.who.int/es/health-topics/coronavirus#tab=tab_1.
- [14] Shaoqing Ren Kaiming He, Xiangyu Zhang and Jian Sun;. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015.
- [15] Lall B Kaul P. *Theoretical Characterization of Deep Neural Networks*, pages 25–63. Springer International Publishing, Cham, 2020. ISBN 978-3-030-31756-0. doi: 10.1007/978-3-030-31756-0_2. URL https://doi.org/10.1007/978-3-030-31756-0_2.
- [16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. doi: 10.1145/3065386.
- [17] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen A. W. M. van der Laak, Bram van Ginneken, and Clara I. Sánchez. A survey on deep learning in medical image analysis, Feb 2017. URL <https://arxiv.org/abs/1702.05747v1>.
- [18] Jose Martinez. Gradiente descendiente para aprendizaje automático, Sep 2020. URL https://www.iartificial.net/gradiente-descendiente-para-aprendizaje-automatico/#%C2%BFQue_es_el_Gradiente.
- [19] Jesús Martínez. ¿qué es un optimizador y para qué se usa en deep learning?, Jul 2020. URL <https://datasmarts.net/es/que-es-un-optimizador-y-para-que-se-usa-en-deep-learning/#>.
- [20] Alma E. Martínez Licona and John Goddard Close. Definición de una red neuronal para clasificación por medio de un programa evolutivo. *Mexican Journal of Biomedical Engineering*, 22(1):8, Mar. 2001. URL <http://rmib.com.mx/index.php/rmib/article/view/417>.
- [21] IBM SPSS MODELER, Aug 2021. URL <https://www.ibm.com/docs/es/spss-modeler/saas?topic=networks-neural-model>.
- [22] Wendy Morales Castro and Rafael Guzman Cabrera. Tuberculosis: Diagnosis by image processing. *Computación y Sistemas*, 24(2), 2020. doi: 10.13053/cys-24-2-3284.
- [23] Yani Muhamad, Irawan Budhi, and Casi Setiningsih. Application of transfer learning using convolutional neural network method for early detection of terry’s nail. *Journal of Physics: Conference Series*, 1201(1): 012052, 2019. doi: 10.1088/1742-6596/1201/1/012052.
- [24] Daniel Nieto Bonilla. *Sistema de reconocimiento de kanjis japoneses basado en procesamiento digital de imágenes aplicado a dispositivos móviles*. PhD thesis, Universidad de las Américas Puebla, 2010.
- [25] Tawsifur Rahman. Covid-19 radiography database, Mar 2022. URL <https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database/data>.
- [26] Mathias Rieder and Richard Verbeet. *Hamburg International Conference of Logistics*. Cognitive Concepts to Improve Collaboration of Human Operators and Service Robots in Picking Processes, 2019.
- [27] Andrés Catalán Urzúa. Técnicas de procesamiento digital de imágenes. *Revista de Marina Año CXXXIC*, 136(969), May 2019. doi: 0719-4129. URL <https://revistamarina.cl/es/articulo/tecnicas-de-procesamiento-digital-de-imagenes#:~:text=El%20procesamiento%20digital%20de%20im%C3%A1genes,b%C3%BAsqueda%20de%20informaci%C3%B3n%20en%20ellas>.

-
- [28] Juan Francisco Vallalta. Aprendizaje supervisado y no supervisado, Aug 2019. URL <https://healthdataminer.com/data-mining/aprendizaje-supervisado-y-no-supervisado/#:~:text=El20aprendizaje20supervisado20supone20que,de20datos20no20etiquetados20previamente>.
- [29] Vaibhav Verdhan. *Image Classification Using LeNet*, pages 67–101. Apress, Berkeley, CA, 2021. ISBN 978-1-4842-6616-8. doi: 10.1007/978-1-4842-6616-8_3. URL https://doi.org/10.1007/978-1-4842-6616-8_3.
- [30] Frank Rosenblatt y Cornell Aeronautical Laboratory. The perceptron : a theory of statistical separability in cognitive systems. *Cornell Aeronautical Laboratory, Buffalo, N.Y.*, 1958.

